

VISIT...

LANZAROTE
Caliente.COM

В.А.Василенко

СПЛАЙН- ФУНКЦИИ:

теория,
алгоритмы,
программы

АКАДЕМИЯ НАУК СССР
СИБИРСКОЕ ОТДЕЛЕНИЕ
ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР

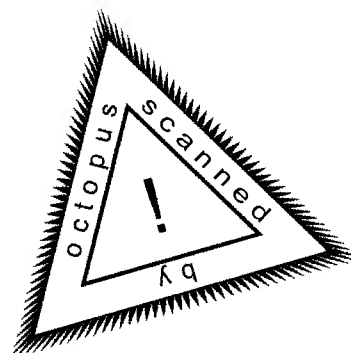
В. А. ВАСИЛЕНКО

СПЛАЙН-ФУНКЦИИ: ТЕОРИЯ, АЛГОРИТМЫ, ПРОГРАММЫ

Ответственный редактор
акад. Г. И. Марчук



ИЗДАТЕЛЬСТВО «НАУКА»
СИБИРСКОЕ ОТДЕЛЕНИЕ
Новосибирск · 1983



УДК 519.6

В а с и л е н к о В. А. Сплайн-функции: теория, ритмы, программы.— Новосибирск: Наука, 1983.

Излагаются основы вариационной теории сплайн-функций. Наряду с теоретическими вопросами, касающимися существования, единственности, сходимости решений задач сплайн-приближений в функциональных пространствах, подробно рассматриваются наиболее важные сплайновые конструкции с точки зрения практического построения, выводятся и анализируются расчетные формулы, обсуждаются вопросы организации вычислений и программ. Описывается программный комплекс, реализующий большинство рассмотренных алгоритмов, приводятся тексты программ и тестовые таблицы.

Для научных работников и инженеров, применяющих методы сплайнов.

Ил. 14. Табл. 49. Библиогр. 63.

ПРЕДИСЛОВИЕ

Необходимость приближенного представления функций при решении конкретных задач — проблема, хорошо знакомая каждому математику или инженеру, работающему в области приложений. Она возникает обычно по двум причинам. Первая предполагает наличие аналитического, но трудновычислимого объекта, который следует заменить более простым, быть может проиграв при этом в точности, но выиграв в экономичности. Вторая причина состоит в том, что исходные данные дискретны, а задача может требовать некоторого функционального представления кривой или поверхности.

Классический аппарат решения таких задач — теория полиномиальных и дробно-рациональных приближений, развитая в работах К. Вейерштрасса, П. Л. Чебышева, С. Н. Бернштейна и др. Однако аппарат полиномиальных приближений мало пригоден для аппроксимации функций с конечной, притом небольшой, гладкостью, такие объекты чаще всего встречаются в приложениях. Это обстоятельство и привело к необходимости введения сплайнов.

Первые сплайн-функции, предложенные И. Шёнбергом (1946), были «склеены» из кусков кубических многочленов. В дальнейшем эта конструкция модифицировалась, повышалась степень многочленов, изменялись краевые условия, но идея оставалась неизменной. Следующий существенный шаг в теории сплайнов — обнаруженное Дж. Холлидеем (1957) свойство, связавшее кубические сплайны Шёнберга с решением вариационной задачи о минимуме квадратичного функционала потенциальной энергии изгиба упругого стержня.

Это обстоятельство вызвало значительный интерес, и в течение последующих лет появилось большое число работ, где в зависимости от конкретных требований модифицировался вариационный энергетический функционал — ведь каждой физической задаче присущи свои энер-

гетические принципы, свой функционал потенциальной энергии, минимум которого обеспечивает наиболее разумное решение задачи аппроксимации. С течением времени от решения задач интерполяции, когда в узлах сетки заданы значения сплайна, исследователи стали переходить к решению задач, где в узлах задавались производные (эрмитовы сплайн-аппроксимации) и сложные дифференциальные выражения (коллокационные методы). Параллельно с этим направлением развивалась теория сглаживания, когда в вариационном функционале комбинировались функционал энергии и интерполяционные условия с некоторыми весами, что позволяло строить «компромиссные» сплайн-аппроксимации с учетом обусловленных экспериментом помех во входных данных. На этом этапе в каждом конкретном случае изучались вопросы существования, единственности, сходимости сплайнов, алгоритмы построения, которые становились все более сложными, а формулы все более громоздкими.

Именно в этот момент появились работы М. Атья (1965—1966), Ф. М. Анселона, П.-Ж. Лорана (1968), в которых было дано общее определение сплайна как элемента абстрактного гильбертова пространства, доказаны теоремы существования и единственности сплайнов, даны алгоритмы для построения сплайнов. С этого времени стала очевидной связь между теорией сплайнов и теорией регуляризации некорректных задач — на первый взгляд, совсем другим направлением, развивавшимся школой советских математиков во главе с А. Н. Тихоновым. Так были заложены основы современной вариационной теории сплайнов, изложению которой в общей форме и на примерах посвящена эта книга.

Структура данной монографии в какой-то мере постулирована в ее названии: теория, алгоритмы, программы. Каждая глава (за исключением гл. 6) построена по дедуктивному принципу — от сложного к простому. Изложение всегда начинается с формулировки и доказательства общих результатов для абстрактных сплайнов, затем рассматривается серия самых важных, общеупотребительных сплайнов и на их примере подробно прорабатываются общие результаты, выводятся расчетные формулы, анализируются вычислительные аспекты, даются конкретные рекомендации по организации программ и численной реализации. Наконец, в гл. 6 приводятся инструкции и тестовый материал к конкретным комплектам

программ. В приложениях расположены тексты программ на языке АЛГОЛ-60 и тестовые таблицы.

Остановимся подробнее на содержании каждой из глав. В гл. 1 дается определение интерполяционных и сглаживающих сплайнов, доказываются теоремы существования и единственности, рассматриваются примеры сплайнов. Вторая глава посвящена изложению двух алгоритмов для построения сплайнов: алгоритма Анселона — Лорана, который детально рассматривается для полиномиальных сплайнов, и алгоритма с использованием функций Грина, который иллюстрируется на примере сплайн-аппроксимации в многомерной области на хаотической сетке. В гл. 3 изложено применение метода конечных элементов для приближенного построения сложных неполиномиальных сплайн-функций, как интерполяционных, так и сглаживающих. Теория иллюстрируется на примере двумерной задачи с хаотической сеткой. Гл. 4 посвящена вопросам сходимости сплайнов. В ней доказана общая теорема сходимости, применение которой иллюстрируется на примере полиномиальных сплайнов, а также сплайнов двух переменных на сгущающейся хаотической сетке. Здесь же доказана теорема сходимости последовательности сглаживающих сплайнов к интерполяционному и рассмотрена проблема выбора параметра сглаживания. Гл. 5 несколько выпадает из общего изложения. Она посвящена проблеме сглаживания данных не с помощью сплайнов, а с помощью осредняющих операторов типа свертки. Общая теория построения ядер таких операторов с заданными свойствами иллюстрируется на примере кусочно-постоянных осреднений, сохраняющих тригонометрические, экспоненциальные и полиномиальные функции. Гл. 6 представляет собой описание библиотеки программ аппроксимации и сглаживания функций одной и многих переменных. Эта глава чрезвычайно самостоятельна. Если читатель по какой-то причине не желает знакомиться с теоретическими аспектами алгоритмов, реализованных в библиотеке, а хочет лишь воспользоваться готовыми программами, это не вызовет затруднений. Подробные инструкции, детальный тестовый материал дают возможность не только работать с библиотекой на языке АЛГОЛ-60, но, при необходимости, переработать ее на другой язык программирования. Следует отметить, однако, что знакомство с основным материалом книги позволит не только использовать готовые сплайны, но и строить другие сплайновые конструкции, наибо-

лее приспособленные для решения конкретной задачи. Именно в этом автор видит основную цель книги.

Практически все доказательства теорем приведены полностью. Для их понимания и чтения книги в полном объеме необходимо располагать сведениями из функционального анализа и линейной алгебры в рамках университетского курса. Ссылки на такую литературу расположены непосредственно в тексте. Это не относится к литературе по теории сплайнов, ссылки на которую в основном тексте не требуются. В конце книги расположены комментарии к главам, в которых подробно указаны соответствующие источники. Без анализа доказательств, только с точки зрения применения алгоритмов и программ, монография доступна более широкому кругу инженерно-технических работников.

Предлагаемая книга дополняет переведенные на русский язык монографии Дж. Альберга, Э. Нильсона и Дж. Уолша «Теория сплайнов и ее приложения» (1972), П.-Ж. Лорана «Аппроксимация и оптимизация» (1975), монографии С. Б. Стечкина, Ю. Н. Субботина «Сплайны в вычислительной математике» (1976), В. И. Бердышева, Ю. Н. Субботина «Численные методы приближения функций» (1979), Ю. С. Завьялова, Б. И. Квасова, В. Л. Мирошниченко «Методы сплайн-функций» (1980).

Работы по теории сплайн-приближений и созданию соответствующего программного обеспечения, проводившиеся в Вычислительном центре Сибирского отделения АН СССР и составившие основу этой книги, пользовались неизменной поддержкой академика Г. И. Марчука и во многом инициировались им, за что автор ему искренне признателен.

Автор считает приятным долгом поблагодарить А. В. Ковалкова, М. В. Зюзина и А. В. Бежаева за ряд полезных замечаний по тексту рукописи и содействие при подготовке программного материала, д-ра физ.-мат. наук Ю. Н. Субботина и канд. техн. наук Л. А. Оганесяна, которые отметили и помогли устранить некоторые неточности. Особую признательность автор выражает канд. физ.-мат. наук В. И. Агошкову, детальные обсуждения с которым помогли усовершенствованию книги в целом и сделали ее более доступной для чтения.

ВВЕДЕНИЕ

Абстрактное понятие сплайна, которым мы будем оперировать в рамках всей книги, исторически возникло из очень простой постановки интерполяционной задачи. Пусть на отрезке $[a, b]$ задана сетка

$$a = x_1 < x_2 < \dots < x_N = b,$$

в узлах которой известны значения $z_1, z_2, \dots, z_N$. На отрезке рассмотрим класс $W_2^2[a, b]$ функций, 2-я производная которых суммируема с квадратом на $[a, b]$, $\int_a^b f''^2(x) dx < \infty$ для любой функции $f(x)$ из $W_2^2[a, b]$. В этом классе будем искать функцию $\sigma(x)$, принимающую в узлах заданные значения,

$$\sigma(x_i) = z_i, \quad i = 1, 2, \dots, N.$$

Разумеется, таких функций много. Среди всех функций класса W_2^2 , проходящих через точки (x_i, z_i) , выберем такую, чтобы

$$\|\sigma''\|_{L_2}^2 = \int_a^b [\sigma''(x)]^2 dx = \min.$$

Величину $\|\sigma''\|_{L_2}^2$ легко трактовать физически. Это потенциальная энергия изгиба упругого однородного стержня, закрепленного в точках плоскости (x_i, z_i) , $i = \overline{1, N}$. Минимум потенциальной энергии, как оказалось, достигается на функции определенного типа, так называемом кубическом сплайне Шёнберга [1], который устроен следующим образом: в узлах x_i он принимает значение z_i , на каждом интервале $[x_i, x_{i+1}]$ является кубическим многочленом, имеет в точках x_i две непрерывные производные.

Кубические сплайны обладают многими полезными свойствами: они хорошо аппроксимируют не слишком гладкие функции, легко конструируются и т. д. Однако не следует забывать, что они описывают именно положение упругого стержня, а в реальной физической задаче совсем не обязательно достигается минимум потенциальной энергии изгиба. Возможно, естествен совсем другой функционал энергии. Это обстоятельство привело к обобщению понятия функционала энергии, который следует минимизировать. Вместо $\|\sigma''\|_{L_2}^2$ стали рассматривать абстрактную величину

$$E(\sigma) = \|T\sigma\|_Y^2,$$

где σ — элемент некоторого функционального пространства X , T — линейный ограниченный оператор, действующий в другое функциональное пространство Y . В частном случае для кубических сплайнов Шёнберга $X = W_2^2[a, b]$, $T = \frac{d^2}{dx^2}$, $Y = L_2[a, b]$.

Далее, для сплайнов Шёнберга в узлах x_i заданы значения z_i . Однако в реальной ситуации относительно функции, которую мы хотим восполнить, могут быть известны совсем другие величины, например наклоны, кривизны, интегралы, следы функции на кривых и т. д. Поэтому простейшее интерполяционное условие $\sigma(x_i) = z_i$, $i = \overline{1, N}$ следует заменить более общим

$$A\sigma = z,$$

где A — некий линейный ограниченный оператор, действующий из X , в котором расположено σ , в пространство Z , быть может, довольно сложной структуры. Эта структура зависит от характера «измерений», которые проделаны над реальным физическим объектом. В случае сплайнов Шёнберга «измерены» значения функции в узлах сетки.

Итак, возникает уравнение $A\sigma = z$, и нужно отыскать такое решение, на котором величина $\|T\sigma\|_Y^2$ минимальна. Это решение и есть абстрактный сплайн. Конечно, измерения сделаны с некоторыми погрешностями и могут быть противоречивыми. В этом случае изучаются так называемые «сглаживающие» сплайны, о которых мы будем вести речь весьма подробно.

Линейность оператора T гарантирует нам квадратичность функционала энергии, а линейность оператора A предполагает линейность измерений. Эти предположения упрощают ситуацию и позволяют построить содержательную теорию, с которой мы Вас познакомим.

При чтении монографии, которая в основном написана на языке функционального анализа, автор рекомендует читателям для лучшего понимания абстрактных результатов возвращаться к простейшей ситуации — сплайнам Шёнберга, когда оператор T — двукратное дифференцирование, а оператор A — оператор следа функции на сетку $x_1, x_2, \dots, x_N$.

ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ СПЛАЙНОВ

В рамках этой главы будут даны общие определения интерполяционных и сглаживающих сплайнов как элементов гильбертова пространства, доказаны теоремы существования и единственности таких сплайнов, получены так называемые условия ортогональности. Здесь мы первый раз рассмотрим некоторые важные примеры конкретных сплайн-функций, к которым будем возвращаться неоднократно в течение всей книги.

§ 1.1. ПОНЯТИЕ ИНТЕРПОЛЯЦИОННЫХ И СГЛАЖИВАЮЩИХ СПЛАЙНОВ

Пусть X , Y , Z — три гильбертовых пространства с некоторыми, вообще говоря, различными скалярными произведениями. Эти скалярные произведения будем обозначать $(\cdot, \cdot)_X$, $(\cdot, \cdot)_Y$, $(\cdot, \cdot)_Z$, соответствующие нормы — $\|\cdot\|_X$, $\|\cdot\|_Y$, $\|\cdot\|_Z$. Предположим, что из пространства X в пространство Z действует линейный ограниченный оператор $A: X \rightarrow Z$. В пространстве Z задан вектор $z \in Z$. Задача состоит в определении решения $\sigma \in X$ операторного уравнения $A\sigma = z$. Такое уравнение в общем случае может вовсе не иметь решений или иметь много решений. По этой причине введем в рассмотрение линейный ограниченный оператор $T: X \rightarrow Y$, который будем в дальнейшем называть энергетическим оператором. Возможны две ситуации. Первая состоит в том, что уравнение $A\sigma = z$ не противоречиво, т. е. множество

$$A^{-1}(z) = \{x \in X : Ax = z\}, \quad (1.1)$$

называемое полным прообразом элемента z при отображении A , не пусто. В этом случае можно определить интерполяционный сплайн $\sigma \in X$.

Определение 1. Пусть $A^{-1}(z) \neq \emptyset$. Интерполяционным сплайном σ называется элемент пространства X , удов-

летворяющий равенству

$$\|T\sigma\|_Y^2 = \min_{x \in A^{-1}(z)} \|Tx\|_Y^2. \quad (1.2)$$

В общем случае оператор T может иметь ненулевое ядро. Это означает, что величина $\|Tx\|_Y$ является полунормой в пространстве X . Значит, интерполяционный сплайн $\sigma \in X$ — такое решение уравнения $A\sigma = z$, которое доставляет минимум этой полунорме. Такие решения называют обычно T -нормальными [13]. Величину $\|Tx\|_Y^2$ можно трактовать как некий энергетический функционал, который следует минимизировать при выборе решения.

Вторая ситуация состоит в том, что $A^{-1}(z) = \emptyset$, т. е. уравнение $A\sigma = z$ противоречиво. В этом случае введем в рассмотрение вещественный параметр $\alpha > 0$ и построим в X квадратичный функционал

$$\Phi_\alpha(x) = \alpha \|Tx\|_Y^2 + \|Ax - z\|_Z^2. \quad (1.3)$$

В этом функционале с весом $\alpha > 0$ скомбинированы невязка уравнения и энергетический функционал.

Определение 2. Назовем элемент $\sigma_\alpha \in X$ сглаживающим сплайном, если справедливо равенство

$$\Phi_\alpha(\sigma_\alpha) = \min_{x \in X} \Phi_\alpha(x). \quad (1.4)$$

Параметр $\alpha > 0$ будем называть параметром сглаживания.

Задачу сглаживания (1.4) можно ставить и в том случае, когда $A^{-1}(z) \neq \emptyset$. Обычно это вызвано нежеланием точно решать задачу $A\sigma = z$, когда вектор z содержит в себе некоторые измерительные ошибки.

Понятие сглаживающего сплайна в общей постановке эквивалентно понятию регуляризованного решения некоторой некорректно поставленной задачи $A\sigma = z$. Учитывая, однако, что наша основная цель состоит в построении алгоритмов интерполяции и сглаживания данных, в дальнейшем мы будем пользоваться терминологией сплайн-функций: A — некоторый оператор, определяемый структурой произведенных измерений, z — вектор входных данных, результат этих измерений, T — выбранный из физических соображений энергетический оператор, X — конкретное функциональное пространство, состоящее из функций такой степени гладкости, которой обладает реальный измеряемый объект.

§ 1.2. ТЕОРЕМЫ СУЩЕСТВОВАНИЯ И ЕДИНСТВЕННОСТИ

Изучим теперь вопрос о существовании и единственности сплайнов. Через $N(T)$ и $N(A)$ обозначим ядра операторов T и A , т. е. множества

$$\begin{aligned} N(T) &= \{x \in X : Tx = \theta_Y\}, \\ N(A) &= \{x \in X : Ax = \theta_Z\}, \end{aligned} \quad (1.5)$$

где θ_Y и θ_Z — нулевые элементы пространств Y и Z .

Теорема 1.1. Если множество $TN(A)$ замкнуто в пространстве Y и

$$N(A) \cap N(T) = \{\theta_X\}, \quad (1.6)$$

интерполяционный сплайн $\sigma \in X$, являющийся решением задачи

$$\begin{aligned} A\sigma &= z, \\ \|T\sigma\|_Y^2 &= \min, \end{aligned} \quad (1.7)$$

существует и единствен для любого $z \in Z$ такого, что $A^{-1}(z)$ не пусто.

Доказательство. Пусть $A^{-1}(z) \neq \emptyset$. Рассмотрим элемент x из $A^{-1}(z)$. Очевидно, что множество $A^{-1}(z)$ может быть представлено в виде

$$A^{-1}(z) = x + N(A). \quad (1.8)$$

С геометрической точки зрения, $A^{-1}(z)$ — результат сдвига линейного пространства $N(A)$, ядра оператора A , на вектор x , т. е. $A^{-1}(z)$ — это аффинное многообразие в пространстве X . Подействуем на элементы $A^{-1}(z)$ линейным оператором T

$$TA^{-1}(z) = Tx + TN(A). \quad (1.9)$$

Таким образом, $TA^{-1}(z)$ — снова аффинное многообразие, только уже в пространстве Y . Ясно, что в условиях теоремы $TA^{-1}(z)$ замкнуто в Y .

Далее, второе условие из (1.7) представляет собой условие минимального расстояния от точки θ_Y до аффинного многообразия (плоскости) $TA^{-1}(z)$. В силу замкнутости $TA^{-1}(z)$ минимум расстояния реализуется в точке $f \in TA^{-1}(z)$, и такая точка в гильбертовом пространстве Y на этой плоскости только одна. В самом деле, пусть точек минимума две: f_1 и f_2 . Разумеется, $\|f_1\|_Y = \|f_2\|_Y$.

Если f_1 и f_2 принадлежат плоскости $TA^{-1}(z)$, то ей принадлежит и точка $(f_1 + f_2)/2$. Тогда

$$\begin{aligned} \|f_1\|_Y &\leq \|(f_1 + f_2)/2\|_Y \leq (\|f_1\|_Y + \|f_2\|_Y)/2 = \|f_1\|_Y, \\ \|f_2\|_Y &\leq \|(f_1 + f_2)/2\|_Y \leq (\|f_1\|_Y + \|f_2\|_Y)/2 = \|f_2\|_Y. \end{aligned} \quad (1.10)$$

Складывая неравенства (1.10), получаем

$$\|f_1 + f_2\|_Y = \|f_1\|_Y + \|f_2\|_Y. \quad (1.11)$$

Возведем обе части (1.11) в квадрат:

$$\begin{aligned} \|f_1\|_Y^2 + 2(f_1, f_2)_Y + \|f_2\|_Y^2 &= \|f_1\|_Y^2 + 2\|f_1\|_Y \cdot \|f_2\|_Y + \|f_2\|_Y^2 \\ \text{или} \quad (f_1, f_2)_Y &= \|f_1\|_Y \cdot \|f_2\|_Y. \end{aligned} \quad (1.12)$$

Это возможно только при $f_1 = \lambda f_2$, $\lambda \geq 0$. Так как $\|f_1\|_Y = \|f_2\|_Y$, то $\lambda = 1$ и $f_1 = f_2$.

Докажем теперь, что в условиях теоремы существует только один элемент σ такой, что $\sigma \in A^{-1}(z)$ и $T\sigma = f$. Предположим, что таких элементов два, σ_1 и σ_2 . Тогда

$$\begin{aligned} T\sigma_1 &= f, \quad A\sigma_1 = z, \\ T\sigma_2 &= f, \quad A\sigma_2 = z. \end{aligned} \quad (1.13)$$

Рассмотрим разность $g = \sigma_1 - \sigma_2$. Ясно, что

$$Tg = \theta_Y, \quad Ag = \theta_Z, \quad (1.14)$$

другими словами, g принадлежит множеству $N(T) \cap N(A)$, и, следовательно, по условию теоремы $g = \theta_X$, $\sigma_1 = \sigma_2$. Теорема доказана.

Возникает вопрос: какие требования следует наложить на операторы T и A , чтобы множество $TN(A)$ было замкнуто?

Теорема 1.2. Если образ $R(T) = TX$ оператора T замкнут в Y , а ядро $N(T)$ конечномерно, линейное многообразие $TN(A)$ замкнуто в Y .

Доказательство. Рассмотрим последовательность y_k элементов $TN(A)$, сходящуюся в пространстве Y к y_* . Вектор y_* принадлежит $R(T)$ в силу замкнутости. Следовательно, существует последовательность $x_k \in N(A)$ такая, что $y_k = Tx_k$. Докажем, что последовательность $\{x_k\}_{k \geq 1}$ ограничена. Для этого представим x_k в виде $x_k = x_{k,1} + x_{k,2}$, где $x_{k,1} \in N(T)$, $x_{k,2} \in N(T)^\perp$. Такое разложение однозначно. Ясно, что $Tx_{k,2} = y_k$. Сужение $\tilde{T}$ оператора T на подпространство $N(T)^\perp$ осуществляет непрерывное взаимно-однозначное соответствие между $N(T)^\perp$ и $R(T)$. По теореме Банаха об обратном операторе [9], опе-

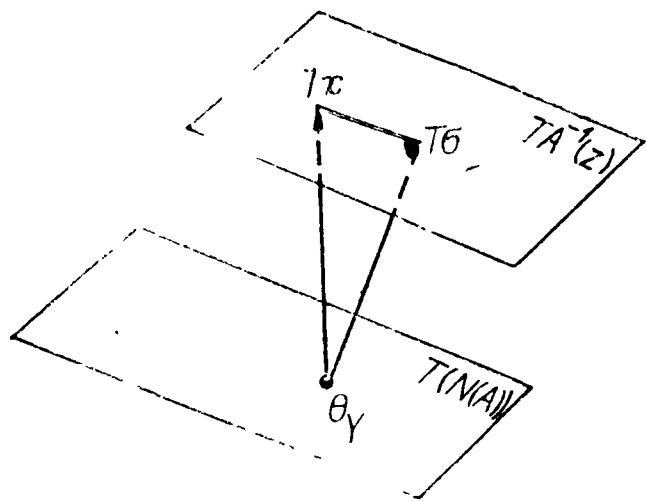


Рис. 1.

ратор T^{-1} ограничен, отсюда

$$\|x_{k,2}\|_X = \|T^{-1}y_k\|_X \leq \|T^{-1}\| \times \|y_k\|_Y. \quad (1.15)$$

Используя ограниченность сходящейся последовательности $\{y_k\}$, получаем ограниченность последовательности $\{x_{k,2}\}$.

Далее, имеем $A(x_{k,1} + x_{k,2}) = \theta_Z$, $Ax_{k,1} = -Ax_{k,2}$. Рассмотрим сужение $\tilde{A}$ оператора A на конечномерное подпространство $N(T)$. Так как $N(T) \cap N(A) = \{\theta_X\}$, $\tilde{A}$ осуществляет взаимно-однозначное соответствие между $N(T)$ и $AN(T)$ и, конечно, $\tilde{A}^{-1}$ ограничен. Следовательно,

$$\|x_{k,1}\| = \|\tilde{A}^{-1}Ax_{k,2}\| \leq \|\tilde{A}^{-1}\| \cdot \|A\| \cdot \|x_{k,2}\| \quad (1.16)$$

и последовательность $\{x_k\}$ ограничена.

Выделим из $\{x_k\}$ слабо сходящуюся последовательность $x_{k'} \xrightarrow{c} x_*$. Тогда $Ax_{k'} \xrightarrow{c} Ax_*$, $Tx_{k'} \xrightarrow{c} Tx_*$. Но мы знаем, что $Ax_{k'} = \theta_Z$ для всех k' , $Tx_{k'} \rightarrow y_*$. Значит, $Ax_* = \theta_Z$, $Tx_* = y_*$, или, другими словами, $y_* \in TN(A)$.

З а м е ч а н и е. В условиях теоремы 1.2 требование конечномерности ядра $N(T)$ оператора T можно заменить требованием замкнутости множества $AN(T)$. Доказательство при этом полностью сохраняется.

Следствие 1. Из геометрической интерпретации задачи (1.7) и свойств гильбертова пространства Y очевидно, что минимум расстояния от точки θ_Y до линейного многообразия $TA^{-1}(z)$ достигается на векторе $T\sigma$, ортогональном линейному пространству $TA^{-1}(\theta_X) = TN(A)$ (рис. 1). Пусть теперь вектор x принадлежит $A^{-1}(z)$. Запишем для прямоугольного треугольника на рис. 1 теорему Пифагора:

$$\|T\sigma\|_Y^2 + \|T(\sigma - x)\|_Y^2 = \|Tx\|_Y^2.$$

Таким образом, для любого x , удовлетворяющего соотношению $Ax = z$, справедливо равенство

$$\|T(\sigma - x)\|_Y^2 = \|Tx\|_Y^2 - \|T\sigma\|_Y^2. \quad (1.17)$$

Это соотношение называют обычно условием ортогональности для интерполяционного сплайна σ . Иногда полезно переписать (1.17) в другой форме:

$$(T\sigma, T(x - \sigma))_Y = 0$$

или для любого $x \in A^{-1}(z)$

$$\|T\sigma\|_Y^2 = (Tx, T\sigma)_Y. \quad (1.18)$$

Сформулируем теорему существования и единственности для решения σ_α задачи сглаживания (1.4).

Введем в рассмотрение пространство $F = Y \times Z$, каждый элемент f которого представляет собой пару $[y, z]$, $y \in Y$, $z \in Z$. Пусть $f_1 = [y_1, z_1]$, $f_2 = [y_2, z_2]$ — два элемента из F . Определим скалярное произведение между ними по формуле

$$(f_1, f_2)_F = ([y_1, z_1], [y_2, z_2])_F = \alpha(y_1, y_2)_Y + (z_1, z_2)_Z. \quad (1.19)$$

Очевидно, что F при этом становится гильбертовым пространством. Определим линейный ограниченный оператор $L: X \rightarrow F$,

$$Lx = [Tx, Ax] \quad (1.20)$$

и вектор $a = [\theta_Y, z]$. Тогда функционал $\Phi_\alpha(x)$ можно переписать в эквивалентной форме

$$\Phi_\alpha(x) = (Lx - a, Lx - a)_F = \|Lx - a\|_F^2. \quad (1.21)$$

В самом деле,

$$\begin{aligned} (Lx - a, Lx - a)_F &= ([Tx, Ax - z], [Tx, Ax - z])_F = \\ &= \alpha\|Tx\|_Y^2 + \|Ax - z\|_Z^2 = \Phi_\alpha(x). \end{aligned}$$

Теорема 1.3. Если образ оператора $L: X \rightarrow F$ замкнут в пространстве F и, кроме того, $N(T) \cap N(A) = \{\theta_X\}$, сглаживающий сплайн σ_α , являющийся решением задачи (1.4), существует и единствен при любом $z \in Z$.

Доказательство. По условию теоремы, образ $R(L)$ оператора L замкнут в F , к тому же оператор L обладает нулевым ядром. Действительно, если $Lx = \theta_F$, это означает, что

$$\begin{aligned} Lx &= [Tx, Ax] = [\theta_Y, \theta_Z], \\ Tx &= \theta_Y, \quad Ax = \theta_Z. \end{aligned}$$

Привлекая условие $N(T) \cap N(A) = \{\theta_X\}$, получаем $x = \theta_X$. Для решения задачи

$$\|L\sigma_\alpha - a\|_F^2 = \min_{x \in X} \|Lx - a\|_F^2 \quad (1.22)$$

необходимо отыскать вектор $L\sigma_\alpha = f \in R(L)$, на котором реализуется минимум расстояния от фиксированного вектора $a \in F$ до линейного пространства $R(L)$ (рис. 2). В си-

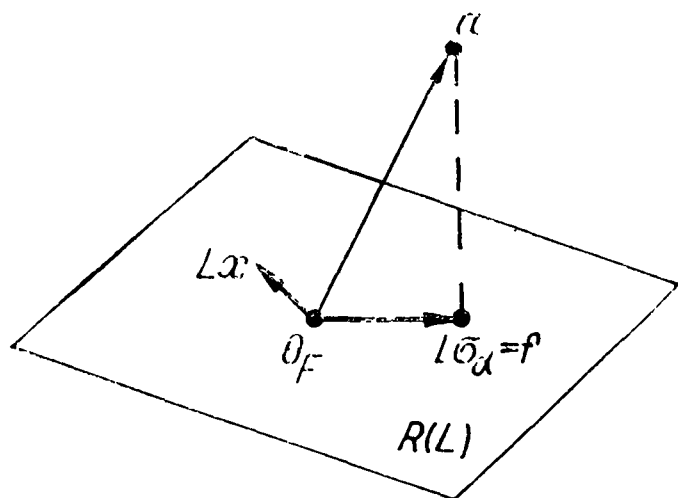


Рис. 2.

лу замкнутости $R(L)$ такой вектор f существует; его единственность доказывается так же, как в теореме 1.1. Оператор L обладает нулевым ядром и, следовательно, однозначно обратим. Уравнение $L\sigma_\alpha = f$ имеет единственное решение $\sigma_\alpha = L^{-1}f$. Теорема доказана.

Следствие. Так как минимум расстояния точки a до $R(L)$ достигается ортогональным проектированием в пространстве F , мы можем для любого вектора $x \in X$ записать

$$(Lx, L\sigma_\alpha - a)_F = 0 \quad (1.23)$$

или в развернутой форме

$$\begin{aligned} (L\sigma_\alpha - a, Lx)_F &= ([Tx, Ax], [T\sigma_\alpha, A\sigma_\alpha - z])_F = \\ &= \alpha(T\sigma_\alpha, Tx)_Y + (A\sigma_\alpha - z, Ax)_Z = 0. \end{aligned}$$

Вычитая из обеих частей $(A\sigma_\alpha - z, z)$, получим, что для любого $x \in X$ справедливо равенство

$$\alpha(T\sigma_\alpha, Tx)_Y + (A\sigma_\alpha - z, Ax - z)_Z = -(A\sigma_\alpha - z, z)_Z. \quad (1.24)$$

Это равенство будем называть условием ортогональности для сглаживающих сплайнов. Отметим, что правая часть (1.24) не зависит от вектора x .

Пусть условия теорем 1.1, 1.3 для операторов T и A выполнены и мы по вектору $z \in Z$ нашли интерполяционный сплайн σ и сглаживающий сплайн σ_α . Справедлива

Теорема 1.4. Сглаживающий сплайн σ_α является некоторым интерполяционным сплайном с вектором входных данных z_α .

Доказательство. Предположим, что сглаживающий сплайн σ_α найден. Вычислим вектор $z_\alpha = A\sigma_\alpha$ и сформулируем задачу отыскания интерполяционного сплайна $\hat{\sigma}$:

$$\|\hat{T}\hat{\sigma}\|_Y^2 = \min_{x \in A^{-1}(z_\alpha)} \|Tx\|_Y^2. \quad (1.25)$$

Эта задача по теореме 1.1 имеет единственное решение $\hat{\sigma}$, так как множество $A^{-1}(z_\alpha)$ не пусто, оно содержит, например, элемент σ_α . Очевидно,

$$\Phi_\alpha(\sigma_\alpha) \leq \Phi_\alpha(\hat{\sigma}). \quad (1.26)$$

С другой стороны,

$$\begin{aligned}\Phi_\alpha(\hat{\sigma}) &= \alpha \|T\hat{\sigma}\|_Y^2 + \|A\hat{\sigma} - z\|_Z^2 = \alpha \|T\hat{\sigma}\|_Y^2 + \|A\sigma_\alpha - z\|_Z^2 \leq \\ &\leq \alpha \|T\sigma_\alpha\|_Y^2 + \|A\sigma_\alpha - z\|_Z^2 = \Phi_\alpha(\sigma_\alpha).\end{aligned}$$

Окончательно $\Phi_\alpha(\sigma_\alpha) = \Phi_\alpha(\hat{\sigma})$, и в силу единственности решения задачи сглаживания (1.4) имеем $\sigma_\alpha = \hat{\sigma}$. Таким образом, σ_α является решением интерполяционной задачи (1.25) с вектором входных данных z_α . Теорема доказана.

Итак, найдены достаточные условия существования и единственности интерполяционных и сглаживающих сплайнов. При сглаживании допускается противоречивость исходной информации, т. е. пустота множества $A^{-1}(z)$.

Пусть теперь вектор входных данных z пробегает множество Z^+ в пространстве Z , для элементов которого $A^{-1}(z)$ не пусто. Очевидно, что Z^+ линейное в Z множество. Пространство решений интерполяционных задач вида (1.2) будем называть пространством интерполяционных сплайнов и обозначать $Sr(T, A)$. Как мы выяснили, согласно теореме 1.4, всевозможные сглаживающие сплайны σ_α для всех $\alpha > 0$ лежат в том же пространстве $Sr(T, A)$.

§ 1.3. ПРИМЕРЫ ЗАДАЧ ИНТЕРПОЛЯЦИИ И СГЛАЖИВАНИЯ

Приведем несколько задач интерполирования и сглаживания данных и сформулируем условия, гарантирующие существование и единственность решений этих задач.

Рассмотрим отрезок $[a, b]$ вещественной оси с нанесенной на него сеткой $a = t_1 < t_2 < \dots < t_N = b$. В узлах сетки заданы значения $z_1, z_2, \dots, z_N$. Пусть $X = W_2^n[a, b]$ — пространство Соболева, состоящее из функций, n -я обобщенная производная которых суммируема с квадратом на $[a, b]$, $n \geq 1$, $Y = L_2[a, b]$ — пространство суммируемых с квадратом функций, $T = \frac{d^n}{dt^n}$ — n -кратное дифференцирование, $A : W_2^n \rightarrow E_N$ — оператор следа функции на сетку,

$$Au = (u(t_1), u(t_2), \dots, u(t_N)). \quad (1.27)$$

Положим $Z = E_N$ — N -мерное евклидово пространство. Сформулируем две задачи.

Задача 1. Найти интерполяционный сплайн $\sigma(t) \in W_2^n[a, b]$ такой, что

$$1) \sigma(t_i) = z_i, \quad i = \overline{1, N};$$

$$2) \left\| \frac{d^n \sigma}{dt^n} \right\|_{L_2}^2 = \min. \quad (1.28)$$

Задача 2. При $\alpha > 0$ найти сглаживающий сплайн $\sigma_\alpha(t) \in W_2^n[a, b]$ такой, что

$$\Phi_\alpha(\sigma_\alpha) = \alpha \left\| \frac{d^n \sigma_\alpha}{dt^n} \right\|_{L_2}^2 + \|A\sigma_\alpha - z\|_{E_N}^2 = \min. \quad (1.29)$$

Справедлива

Теорема 1.5. Сплайны $\sigma(t)$ и $\sigma_\alpha(t)$, являющиеся решениями задач (1.28) и (1.29), существуют и единственны, если количество узлов сетки N превышает или равно порядку дифференцирования n .

Доказательство. Отметим сразу, что образы операторов T и A в задачах (1.28), (1.29) замкнуты. В са-

мом деле, образ оператора $T = \frac{d^n}{dt^n} : W_2^n \rightarrow L_2$ совпадает

со всем пространством L_2 и поэтому замкнут. Образ оператора A является линейным подпространством конечномерного пространства E_N и, следовательно, замкнут. Далее, $A^{-1}(z)$ — непустое множество для всех $z = [z_1, z_2, \dots, z_N]$, так как для любого вектора z существует функция класса $W_2^n[a, b]$, принимающая в узлах сетки t_i значения z_i . Например, это интерполяционный многочлен Лагранжа степени $N - 1$.

Ядро $N(A)$ оператора A состоит из функций класса W_2^n , обращающихся в нуль в узлах сетки $t_1, t_2, \dots, t_N$. Ядро $N(T)$ состоит из многочленов степени, меньшей n . Пересечение $N(A)$ и $N(T)$ при $N \geq n$ содержит, таким образом, только тождественно нулевую функцию, так как нетривиальный многочлен степени $n - 1$ не может иметь более $n - 1$ корней.

Используя теперь теоремы 1.1, 1.2, 1.3, получаем требуемое утверждение. Теорема доказана.

Сохраняя обозначения, заменим в задачах 1, 2 оператор A следа функции на сетку другим. Пусть нам известны не значения функции в узлах, а средние интегральные значения по отрезкам сетки:

$$Ax(t) = \left[\frac{1}{t_2 - t_1} \int_{t_1}^{t_2} x(t) dt, \right. \\ \left. \frac{1}{t_3 - t_2} \int_{t_2}^{t_3} x(t) dt, \dots, \frac{1}{t_N - t_{N-1}} \int_{t_{N-1}}^{t_N} x(t) dt \right]. \quad (1.30)$$

Теорема 1.6. Интерполяционная и сглаживающая задачи (1.28), (1.29) с оператором A , определенным по формуле (1.30), имеют единственное решение при $N \geq n + 1$, $n \geq 0$.

Доказательство. Ядро оператора A , определенного формулой (1.30), состоит из функций класса W_2^n , интегральные средние от которых по отрезкам сетки $[t_i, t_{i+1}]$ равны нулю. Таким образом, любой полином степени $n - 1$ ^{*}), входящий в $N(A) \cap N(T)$, имеет внутри каждого интервала сетки корень. В противном случае нетривиальный полином сохранял бы знак на отрезке и не мог бы давать среднее интегральное значение, равное 0. Отрезков сетки — $N - 1$, корней — тоже $N - 1$, и при $N \geq n + 1$ $N(T) \cap N(A)$ состоит только из нулевого многочлена. Проверка остальных условий применения общих теорем 1.1—1.3 делается так же, как при доказательстве теоремы 1.5.

Задача 3. Рассмотрим единичную окружность на плоскости с центром в точке 0. На окружности заданы несовпадающие точки $t_1, t_2, \dots, t_N$, определяемые углами $\varphi_1, \varphi_2, \dots, \varphi_N$, $0 \leq \varphi_1 < \varphi_2 < \dots < \varphi_N < 2\pi$. Положим $X = W_2^{2m+1}[0, 2\pi)$ — соболевское пространство, состоящее из периодических функций, $Y = L_2[0, 2\pi)$ — тоже из периодических с периодом 2π функций. Определим оператор T формулой

$$T = \Delta_m = D(D^2 + I)(D^2 + 2^2 I) \dots (D^2 + m^2 I),$$

где D — оператор однократного дифференцирования, I — тождественный оператор. Пусть $A : W_2^{2m+1} \rightarrow E_N$ — оператор следа функции на сетку $\varphi_1, \varphi_2, \dots, \varphi_N$. Поставим задачу отыскания периодической функции $\sigma(\varphi) \in W_2^{2m+1}[0, 2\pi)$ из условий

^{*}) При $n = 0$ оператор T является тождественным и имеет нулевое ядро, т. е. $N(A) \cap N(T) = \{\theta_x\}$.

$$\sigma(\varphi_i) = z_i, \quad i = \overline{1, N};$$

$$\int_0^{2\pi} [(\Delta_m \sigma)(\varphi)]^2 d\varphi = \min. \quad (1.31)$$

Теорема 1.7. Решение задачи (1.31) существует и единственно при $N \geq 2m + 1$.

Доказательство. Ядро $N(\Delta_m)$ оператора Δ_m состоит из тригонометрических многочленов вида

$$a_0 + \sum_{\kappa=1}^m (b_\kappa \cos \kappa t + c_\kappa \sin \kappa t). \quad (1.32)$$

Функция вида (1.32) не может иметь на отрезке $[0, 2\pi)$ свыше $2m$ корней. Это и означает, что $N(\Delta_m) \cap N(A) = \{0\}$ при $N \geq 2m + 1$. Оператор Δ_m имеет замкнутый образ, совпадающий со всем пространством $L_2[0, 2\pi)$ периодических с периодом 2π функций. Теорема доказана.

Задача 4. Пусть Ω — некоторая ограниченная область в R^n . В области Ω задан конечный набор несовпадающих точек $P_1, P_2, \dots, P_N$. Введем в рассмотрение оператор обобщенных градиентов порядка m по формуле

$$Tu = \nabla^m u = \left[\sqrt{\frac{m!}{\alpha!}} D^\alpha u, |\alpha| = m \right], \quad (1.33)$$

где $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ — мультииндекс, $\alpha! = \alpha_1! \alpha_2! \dots \alpha_n!$, $|\alpha| = \sum_{i=1}^n \alpha_i$. Оператор $T = \nabla^m$ осуществляет отображение

из пространства $X = W_2^m(\Omega)$ функций, всевозможные частные производные которых до суммарного порядка m включительно интегрируемы с квадратом. Оператор ∇^m действует из $W_2^m(\Omega)$ в пространство обобщенных градиентов, которое представляет собой замкнутое подпространство в пространстве $Y = \prod_{i=1}^R L_2(\Omega)$, где R — количество

различных мультииндексов α , для которых $|\alpha| = m$. Напомним, что замкнутость пространства обобщенных градиентов $\nabla^m W_2^m(\Omega)$ может быть гарантирована [9] для весьма широкого класса областей Ω . Достаточно потребовать, чтобы область Ω была объединением конечного количества областей Ω_i , звездных относительно некоторых шаров. Напомним, что область называется звездной относительно точки, если ее граница полностью «обозревается» из этой точки, и соответственно звездной относи-

тельно некоторого шара, если граница обозревается из всех точек этого шара.

Будем в дальнейшем предполагать, что область Ω именно такова и, кроме того, $m > n/2$. Это условие гарантирует вложение [9] пространства $W_2^m(\Omega)$ в пространство непрерывных функций $C(\Omega)$, в свою очередь, это означает, что оператор следа

$$Au = [u(P_1), u(P_2), \dots, u(P_N)] \quad (1.34)$$

функций $u \in W_2^m(\Omega)$ на сетку $P_1, P_2, \dots, P_N$ является линейным ограниченным оператором из $W_2^m(\Omega)$ в евклидово пространство E_N .

Поставим задачу отыскания интерполяционных сплайн-функций $\sigma(P) \in W_2^m(\Omega)$ из условий

$$\sigma(P_i) = z_i, \quad i = \overline{1, N}; \quad (1.35)$$

$$\|\nabla^m \sigma\|_Y^2 = \sum_{|\alpha|=m} \frac{m!}{\alpha!} \int_{\Omega} (D^\alpha \sigma)^2(P) d\Omega = \min. \quad (1.36)$$

Теорема 1.8. Пусть $r = \frac{(n+m-1)!}{n!(m-1)!}$. Если среди точек $P_1, P_2, \dots, P_N$ найдется r точек $P_{i_1}, P_{i_2}, \dots, P_{i_r}$ таких, что возможно однозначно определить многочлен степени $m-1$, принимающий любые заданные значения в этих узлах, задача (1.35), (1.36) имеет единственное решение.

Доказательство. Образ оператора ∇^m замкнут. Поэтому достаточно проверить условие $N(\nabla^m) \cap N(A)$. Ядро $N(\nabla^m)$ оператора ∇^m состоит из всех многочленов степени $m-1$, имеющих вид

$$\sum_{|\alpha| \leq m-1} a_\alpha t^\alpha, \quad \alpha = (\alpha_1, \alpha_2, \dots, \alpha_n), \quad t^\alpha = t_1^{\alpha_1} t_2^{\alpha_2} \dots t_n^{\alpha_n}. \quad (1.37)$$

Количество коэффициентов такого многочлена — $r = \frac{(n+m-1)!}{n!(m-1)!}$. $N(\nabla^m) \cap N(A)$ состоит из многочленов вида (1.37), принимающих в узлах $P_1, P_2, \dots, P_N$ нулевые значения. Записав это условие в точках $P_{i_1}, P_{i_2}, \dots, P_{i_r}$, получим

$$\sum_{|\alpha| \leq m-1} a_\alpha P_{i_\kappa}^\alpha = 0, \quad \kappa = \overline{1, r}. \quad (1.38)$$

Ясно, что (1.38) — линейная алгебраическая система с квадратной $r \times r$ матрицей. По условию теоремы существует единственный полином, принимающий в узлах

$P_{i_1}, P_{i_2}, \dots, P_{i_r}$ нулевые значения, а именно нулевой полином. Теорема доказана.

В формулировке теоремы (1.8) можно потребовать, чтобы существовали точки $P_{i_1}, P_{i_2}, \dots, P_{i_r}$, для которых матрица из элементов $P_{i_k}^\alpha$ — неособенная.

Интересен случай $m = n = 2$. Задача приобретает в плоской области Ω вид

$$\sigma(P_i) = z_i, \quad i = \overline{1, N}; \quad (1.39)$$

$$\int_{\Omega} (\sigma_{xx}^2 + 2\sigma_{xy}^2 + \sigma_{yy}^2) d\Omega = \min. \quad (1.40)$$

В этом случае $r = 3!/2!1! = 3$. Для существования и единственности решения задачи (1.39), (1.40) достаточно потребовать, чтобы среди точек $P_1, P_2, \dots, P_N$ нашлись три $P_{i_1}, P_{i_2}, P_{i_3}$ с координатами (x_k, y_k) , $k = 1, 2, 3$, для которых

$$\det \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{pmatrix} \neq 0. \quad (1.41)$$

Это означает, что $P_{i_1}, P_{i_2}, P_{i_3}$ не лежат на одной прямой. К сожалению, для других m, n трудно дать геометрическую трактовку теоремы 1.8.

В задачах 3, 4 ничто не мешает нам вместо задач интерполяции рассматривать задачи сглаживания. Теоремы существования и единственности при этом полностью повторяются.

Глава 2

АНАЛИТИЧЕСКИЕ МЕТОДЫ ПОСТРОЕНИЯ СПЛАЙН-ФУНКЦИЙ

§ 2.1. ОБЩИЙ АЛГОРИТМ ПОСТРОЕНИЯ ИНТЕРПОЛЯЦИОННОГО СПЛАЙНА

Здесь речь пойдет о построении точного (аналитического) решения задачи сплайн-интерполирования

$$\|T\sigma\|_Y^2 = \min_{x \in A^{-1}(z)} \|Tx\|_Y^2 \quad (2.1)$$

при условии, что образ $R(A)$ оператора $A : X \rightarrow Z$ конечно-мерен. Точнее, будем предполагать, что $R(A) = Z = E_n$ — n -мерное векторное пространство с обычным скалярным произведением векторов $x = (x_1, x_2, \dots, x_n)$ и $y = (y_1, y_2, \dots, y_n)$:

$$(x, y)_{E_n} = \sum_{i=1}^n x_i y_i. \quad (2.2)$$

В этом случае действие оператора $A : X \rightarrow E_n$ эквивалентно действию n линейно-независимых функционалов k_i , $i = \overline{1, n}$, при этом оператор A связан с функционалами формулой

$$Ax = ((k_1, x)_X, (k_2, x)_X, \dots, (k_n, x)_X). \quad (2.3)$$

Относительно операторов задачи T и A будем предполагать выполнение стандартных (см. гл. 1) требований, гарантирующих существование и единственность сплайна σ . Задача (2.1) становится, таким образом, эквивалентной задаче

$$(k_i, \sigma)_X = r_i, \quad i = \overline{1, n};$$

$$\|T\sigma\|_Y^2 = \min. \quad (2.4)$$

Здесь r_i — компоненты фиксированного вектора $\bar{r} = (r_1, r_2, \dots, r_n)$.

Обозначим через K линейную оболочку системы элементов $k_1, k_2, \dots, k_n$. Очевидно, что ядро $N(A)$ оператора A и линейное пространство K связаны соотношением

$$N(A) = K^\perp,$$

где $K^\perp$ — ортогональное дополнение K (символ $\perp$ будет в дальнейшем обозначать ортогональное дополнение). Не уменьшая общности, можно в дальнейшем считать, что образ $R(T)$ оператора $T : X \rightarrow Y$ совпадает со всем пространством Y .

Как мы уже знаем [см. (1.17) и рис. 1], вектор $T\sigma$ принадлежит линейному пространству $(TK^\perp)^\perp$. Выясним структуру этого пространства более детально. Для этого введем в рассмотрение оператор $T^* : Y \rightarrow X$, определенный тождеством Лагранжа:

$$(Tx, y)_Y = (x, T^*y)_X, \quad \forall x \in X, \quad \forall y \in Y. \quad (2.5)$$

Отметим сразу, что оператор T^* невырожден, т. е. $N(T^*) = \{\theta_Y\}$. В самом деле, ядро и образ прямого и сопряжен-

ного оператора связаны соотношением

$$N(T^*) = R(T)^\perp = Y^\perp = \theta_Y.$$

Предположим теперь, что оператор T имеет ядро конечной размерности $q < \infty$. Рассмотрим линейное пространство $H = K \cap N(T)^\perp$. В предположениях теоремы существования и единственности $K^\perp \cap N(T) = \{\theta_X\}$. Отсюда следует, что размерность пространства H равна $n - q$, где n — количество функционалов. Далее, утверждается, что

$$(TK^\perp)^\perp = T^{*-1}(H). \quad (2.6)$$

Доказательство этого факта несложно. Формула (2.6) равносильна

$$T^*(TK^\perp)^\perp = H.$$

Пусть $x \in T^*(TK^\perp)^\perp$. Очевидно, что $x \in R(T^*) = N(T)^\perp$. Далее, существует $y \in (TK^\perp)^\perp$ такой, что $x = T^*y$. Можно записать цепочку эквивалентных соотношений:

$$\begin{aligned} [y \in (TK^\perp)^\perp] &\Leftrightarrow [(y, TK^\perp)_Y = 0] \Leftrightarrow \\ &\Leftrightarrow [(T^*y, K^\perp)_X = 0] \Leftrightarrow [x \in K]. \end{aligned}$$

Таким образом, $x \in K \cap N(T)^\perp$. Все преобразования можно повторить в другую сторону.

На основе равенства (2.6) можно описать этапы построения интерполяционного сплайна σ .

Шаг 1. Найти базис пространства $H = K \cap N(T)^\perp$. Для этого необходимо найти линейно-независимые элементы $h_1, h_2, \dots, h_{n-q}$,

$$h_i = \sum_{j=1}^n \alpha_{ij} k_j \quad (2.7)$$

такие, чтобы они были ортогональны $N(T)$. Разумеется, выбор коэффициентов α_{ij} всегда возможен, но не однозначен.

Шаг 2. Найти векторы $f_i = T^{*-1}(h_i)$, $i = \overline{1, n - q}$.

Шаг 3. Как мы знаем, $T\sigma = \sum_{i=1}^{n-q} \lambda_i f_i$, где λ_i — некоторые коэффициенты. Найдем их. Для этого умножим $T\sigma$ скалярно на f_j . Получим

$$\begin{aligned} \sum_{i=1}^{n-q} \lambda_i (f_i, f_j)_Y &= (T\sigma, f_j)_Y = (\sigma, T^*f_j)_X = \\ &= (\sigma, h_j)_X = \left(\sigma, \sum_{i=1}^n \alpha_{ji} k_i \right)_X = \end{aligned}$$

$$= \sum_{i=1}^n \alpha_{ji} (k_i, \sigma)_X = \sum_{i=1}^n \alpha_{ji} r_i. \quad (2.8)$$

Итак, для нахождения неизвестных коэффициентов λ_i следует решить линейную алгебраическую систему с симметричной положительно определенной матрицей Грама:

$$\sum_{j=1}^{n-q} (f_i, f_j)_Y \lambda_j = \sum_{j=1}^n \alpha_{ij} r_j, \quad i = \overline{1, n-q}.$$

Элемент $T\sigma = \sum_{i=1}^{n-q} \lambda_i f_i$ найден.

В матричной форме система приобретает вид

$$A\overline{\lambda} = H\overline{r}, \quad (2.9)$$

где A — матрица Грама элементов $f_1, f_2, \dots, f_{n-q}$, H — прямоугольная $(n-q) \times n$ матрица, составленная из элементов α_{ij} , $j = \overline{1, n}$, $i = \overline{1, n-q}$,

$$\overline{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_{n-q})^T, \quad \overline{r} = (r_1, r_2, \dots, r_n)^T.$$

Шаг 4. Необходимо обратить оператор T на векторе $f = \sum \lambda_i f_i$. Интерполяционный сплайн σ построен.

От выбора коэффициентов α_{ij} зависит структура полученной нами линейной алгебраической системы, в частности, будет ли матрица системы ленточной или нет. Нерациональный выбор α_{ij} может затруднить построение сплайна на практике.

§ 2.2. ОБЩИЙ АЛГОРИТМ ПОСТРОЕНИЯ СГЛАЖИВАЮЩЕГО СПЛАЙНА

Напомним, что при решении задач сглаживания никаких новых типов сплайнов не возникает. Сглаживающий сплайн — интерполяционный, с вектором входных данных $\bar{r}_\alpha$ (см. теорему 1.4, гл. 1).

Пусть сглаживающий сплайн σ_α реализует минимум функционала

$$\Phi_\alpha(x) = \alpha \|Tx\|_Y^2 + \sum_{i=1}^n ((k_i, x)_X - r_i)^2. \quad (2.10)$$

Элемент σ_α принимает на функционалах k_i некоторые значения $(k_i, \sigma_\alpha)_X = r_i^\alpha$. Обозначим через A матрицу Грама элементов $f_1, f_2, \dots, f_{n-q}$, а через H — прямоугольную мат-

рицу, составленную из элементов α_{ij} (см. § 2.1). Если $\bar{r}_\alpha = (r_1^\alpha, r_2^\alpha, \dots, r_n^\alpha)^T$, то, как мы уже знаем из (2.9),

$$A\bar{\lambda}_\alpha = H\bar{r}_\alpha \quad (2.11)$$

и $T\sigma_\alpha = \sum_{i=1}^{n-q} \lambda_i^\alpha f_i$, $\bar{\lambda}_\alpha = (\lambda_1^\alpha, \lambda_2^\alpha, \dots, \lambda_{n-q}^\alpha)^T$. Отсюда легко видеть, что

$$\|T\sigma_\alpha\|_Y^2 = \sum_{i,j=1}^{n-q} \lambda_i^\alpha \lambda_j^\alpha (f_i, f_j)_Y = (A\bar{\lambda}_\alpha, \bar{\lambda}_\alpha); \quad (2.12)$$

функционал $\Phi_\alpha(x)$ на n -мерном пространстве интерполяционных сплайнов может быть записан как функция вектора $\bar{r}_\alpha$:

$$\Phi_\alpha(\bar{r}_\alpha) = \alpha(A\bar{\lambda}_\alpha, \bar{\lambda}_\alpha) + \|\bar{r}_\alpha - r\|^2. \quad (2.13)$$

Здесь скалярное произведение и норма берутся в евклидовом пространстве векторов. Условие минимума квадратичного функционала $\Phi_\alpha(\bar{r}_\alpha)$ имеет вид

$$\frac{\partial \Phi_\alpha(\bar{r}_\alpha)}{\partial r_s^\alpha} = 0, \quad s = 1, 2, \dots, n, \quad (2.14)$$

или подробнее

$$\frac{\partial \Phi_\alpha(\bar{r}_\alpha)}{\partial r_s^\alpha} = \alpha \frac{\partial}{\partial r_s^\alpha} (A\bar{\lambda}_\alpha, \bar{\lambda}_\alpha) + 2(r_s^\alpha - r_s) = 0. \quad (2.15)$$

Но матрица A не зависит от $\bar{r}_\alpha$. Поэтому

$$\begin{aligned} \frac{\partial}{\partial r_s^\alpha} (A\bar{\lambda}_\alpha, \bar{\lambda}_\alpha) &= 2 \left(\frac{\partial (A\bar{\lambda}_\alpha)}{\partial r_s^\alpha}, \bar{\lambda}_\alpha \right) = 2 \left(\frac{\partial (H\bar{r}_\alpha)}{\partial r_s^\alpha}, \bar{\lambda}_\alpha \right) = \\ &= 2 \left(\frac{\partial \bar{r}_\alpha}{\partial r_s^\alpha}, H^* \bar{\lambda}_\alpha \right) = 2 [H^* \bar{\lambda}_\alpha]_s. \end{aligned}$$

Окончательно имеем в векторной форме

$$\alpha H^* \bar{\lambda}_\alpha + \bar{r}_\alpha = \bar{r}. \quad (2.16)$$

Умножая (2.16) слева на H , получаем систему для определения неизвестных коэффициентов вектора $\bar{\lambda}_\alpha$:

$$(A + \alpha H H^*) \bar{\lambda}_\alpha = H \bar{r}. \quad (2.17)$$

Неизвестный вектор $\bar{r}_\alpha$ тоже легко определяется:

$$\bar{r}_\alpha = \bar{r} - \alpha H^* \bar{\lambda}_\alpha. \quad (2.18)$$

Итак, первые два шага построения сглаживающего сплайна те же, что в § 2.1. На шаге 3 нужно решить систему (2.17), т. е. найти $T\sigma_\alpha = \sum_{i=1}^{n-q} \lambda_i^\alpha f_i$ и определить по формуле (2.18) вектор $\bar{r}_\alpha$. Шаг 4 делается так же, как в § 2.1.

§ 2.3. СПЛАЙНЫ НЕЧЕТНОЙ СТЕПЕНИ НА ОТРЕЗКЕ

Рассмотрим отрезок (a, b) , $-\infty < a < b < \infty$, с нанесенной на него сеткой $x_1, x_2, \dots, x_N$,

$$a \leq x_1 < x_2 < \dots < x_N \leq b; \quad (2.19)$$

точки x_1 и x_N не обязательно совпадают с a и b . Положим $X = W_2^p(a, b)$, $p \geq 1$, $Y = L_2(a, b)$; $T = \frac{d^p}{dx^p}$. Пусть в узлах сетки нам заданы значения $r_1, r_2, \dots, r_N$, другими словами, функционалы $k_1, k_2, \dots, k_N$ таковы, что для любой функции $u \in W_2^p$

$$(k_i, u)_{W_2^p} = u(x_i), \quad i = \overline{1, N}. \quad (2.20)$$

Поставим задачу отыскания интерполяционной сплайн-функции $\sigma(x) \in W_2^p(a, b)$ из условий

$$\begin{aligned} \sigma(x_i) &= r_i, \quad i = \overline{1, N}; \\ \int_a^b \left(\frac{d^p \sigma}{dx^p} \right)^2 dx &= \min \end{aligned} \quad (2.21)$$

и задачу отыскания сглаживающей сплайн-функции $\sigma_\alpha(x)$ из условия

$$\alpha \int_a^b \left(\frac{d^p \sigma_\alpha}{dx^p} \right)^2 dx + \sum_{i=1}^N [\sigma_\alpha(x_i) - r_i]^2 = \min. \quad (2.22)$$

Как мы выяснили, задачи (2.21), (2.22) имеют единственное решение при $N \geq p$ (см. § 1.3). В этом параграфе мы построим решения этих задач.

Осуществим последовательно этапы построения сплайна (см. § 2.1, 2.2).

2.3.1. Формирование матрицы H

Ядро оператора $T = \frac{d^p}{dx^p}$ состоит из многочленов степени не выше $p - 1$. Размерность ядра равна p , базис ядра состоит из функций $1, x, x^2, \dots, x^{p-1}$. Будем искать функции $h_1, h_2, \dots, h_{N-p}$ в виде

$$h_i = \sum_{j=i}^{i+p} h_{ij} k_j, \quad i = \overline{1, N-p}. \quad (2.23)$$

Коэффициенты h_{ij} следует (см. § 2.1) выбрать так, чтобы

$$(h_i, x^\kappa)_{W_2^p} = 0, \quad \kappa = 0, \dots, p-1, \quad (2.24)$$

или подробнее

$$\sum_{j=i}^{i+p} h_{ij} x_j^\kappa = 0, \quad \kappa = 0, 1, \dots, p-1. \quad (2.25)$$

Условие (2.25) можно интерпретировать следующим образом: на сеточном шаблоне $x_i, x_{i+1}, \dots, x_{i+p}$ из $p+1$ точек следует построить разностную аппроксимацию оператора $\frac{d^p}{dx^p}$ с порядком аппроксимации $O(h^p)$, т. е. такой раз-

ностный аналог, который «аннулирует» все сеточные многочлены до степени $p-1$. Нам достаточно найти эти коэффициенты с точностью до постоянного множителя.

Для автоматической генерации коэффициентов h_{ij} на произвольной неравномерной сетке очень удобно использовать аппарат разделенных разностей [8]. В самом деле, если рассмотреть на шаблоне $x_i, x_{i+1}, \dots, x_{i+p}$ сеточную функцию, которая равна 1 в одной из точек шаблона и 0 во всех остальных, и вычислить разделенную разность порядка p , то она с точностью до постоянного множителя совпадает с коэффициентом разностного аналога, соответствующим той точке, где сеточная функция была равна 1. Алгоритм, естественно, упрощается, если сетка равномерная.

Матрица H , состоящая из элементов h_{ij} , $i = \overline{1, N-p}$, $j = \overline{1, N}$, имеет, таким образом, ленточную структуру. После перенумерации ее элементов по формуле $\bar{h}_{ij} = h_{i, i+j-1}$ она может быть упакована в прямоугольную матрицу $\bar{H}$ размерности $(N-p) \times (p+1)$. Схематически

это выглядит так:

$$H = \left[\begin{array}{c} \overbrace{\quad\quad\quad}^{P+1} \\ \diagdown \quad \diagdown \quad \diagdown \quad \diagdown \\ \underbrace{\quad\quad\quad}_N \end{array} \right] \left. \vphantom{\begin{array}{c} \overbrace{\quad\quad\quad}^{P+1} \\ \diagdown \quad \diagdown \quad \diagdown \quad \diagdown \\ \underbrace{\quad\quad\quad}_N \end{array}} \right\} N-P, \quad \bar{H} = \left[\begin{array}{c} \overbrace{\quad\quad\quad}^{P+1} \\ \vdots \quad \vdots \quad \vdots \quad \vdots \\ \underbrace{\quad\quad\quad}_{N-P} \end{array} \right] \left. \vphantom{\begin{array}{c} \overbrace{\quad\quad\quad}^{P+1} \\ \vdots \quad \vdots \quad \vdots \quad \vdots \\ \underbrace{\quad\quad\quad}_{N-P} \end{array}} \right\} N-P. \quad (2.26)$$

2.3.2. Обращение оператора T^*

Теперь согласно общей теории мы должны найти функции $f_i(t)$, $i = 1, 2, \dots, N - p$, по формуле

$$f_i(t) = \left(\frac{d^p}{dx^p} \right)^{*^{-1}} h_i(t). \quad (2.27)$$

Следует обратить оператор $\left(\frac{d^p}{dx^p} \right)^*$ на элементе h_i , ортогональном ядру оператора $\left(\frac{d^p}{dx^p} \right)$. Такого рода необходимость будет постоянно возникать при построении различных сплайнов, поэтому мы постараемся ответить на вопрос о способе обращения оператора T^* в несколько более общем случае.

Пусть $T: X \rightarrow L_2$ — некоторый линейный оператор. Предположим, что скалярное произведение в пространстве X определяется формулой

$$(u, v)_X \stackrel{df}{=} (u, v)_{L_2} + (Tu, Tv)_{L_2} \quad (2.28)$$

и нашлась функция $G_+(x - t)$, которая удовлетворяет условию

$$TG_+(x - t) = \delta_t, \quad (2.29)$$

где δ_t — функция Дирака [9], сосредоточенная в точке t . Оператор T действует по отношению к аргументу x , t является параметром *).

Теорема 2.1. Пусть $h(x)$ — некоторая функция из X , ортогональная в X ядру $N(T)$ оператора T . Тогда

$$(T^{*-1}h)(t) = (h(x), G_+(x - t))_X. \quad (2.30)$$

*) Здесь и в дальнейшем выражения, в которые входят G_+ и функции Дирака, понимаются в смысле обобщенных функций.

Доказательство. Пусть $f(t) = (T^*)^{-1}h(t)$, $g(t) = (h(x), G_+(x-t))_X$. Требуемое равенство $f(t) = g(t)$ эквивалентно $h(t) = T^*g(t)$. Умножая обе части этого равенства скалярно в X на произвольную функцию $\mu \in X$, получим

$$\forall \mu \in X (h, \mu)_X \stackrel{?}{=} (T^*g, \mu)_X = (g, T\mu)_{L_2}. \quad (2.31)$$

Если (2.31) справедливо, то справедливо и (2.30). Представим функцию $\mu(x) \in X$ в виде

$$\mu(x) = n(x) + (G_+(x-t), T\mu(t))_{L_2}, \quad (2.32)$$

где $n(x) \in N(T)$. Такое представление возможно, так как

$$T(G_+(x-t), T\mu(t))_{L_2} = (\delta_x, T\mu)_{L_2} = (T\mu)(x), \quad (2.33)$$

следовательно, $(G_+(x-t), T\mu(t))_{L_2}$ отличается от $\mu(x)$ только на элемент ядра оператора T . Тогда

$$\begin{aligned} (h, \mu)_X &= (h(x), (G_+(x-t), T\mu(t))_{L_2})_X = \\ &= (h(x), (G_+(x-t), T\mu(t))_{L_2})_{L_2} + (Th(x), T\mu(x))_{L_2}. \end{aligned} \quad (2.34)$$

С другой стороны,

$$\begin{aligned} (g, T\mu)_{L_2} &= ((h(x), G_+(x-t))_X, T\mu(t))_{L_2} = \\ &= ((h(x), G_+(x-t)) + (Th(x), \delta_t)_{L_2}, T\mu(t))_{L_2} = \\ &= ((h), G_+(x-t))_{L_2}, T\mu(t))_{L_2} + (Th(t), T\mu(t))_{L_2}. \end{aligned} \quad (2.35)$$

Порядок интегрирования в формулах (2.34) и (2.35) можно переставить, и теорема доказана.

Таким образом, если известна функция $G_+(x-t)$, то обращение оператора T^* осуществляется по явной формуле (2.30).

2.3.3. Формирование матрицы A

Для оператора $T = \frac{d^p}{dx^p}$ функция Грина G_+ имеет

вид

$$G_+(x-t) = (x-t)_+^{p-1}/(p-1)!, \quad (2.36)$$

где

$$(E)_+ = \begin{cases} E, & E \geq 0; \\ 0, & E < 0. \end{cases} \quad (2.37)$$

Следовательно, на основании (2.30)

$$f_i(t) = \left(\sum_{j=i}^{i+p} h_{ij} k_j, (x-t)_+^{p-1}/(p-1)! \right)_{W_2^p}. \quad (2.38)$$

С учетом (2.20) имеем

$$f_i(t) = \sum_{j=i}^{i+p} h_{ij} (x_j - t)_+^{p-1}/(p-1)!, \quad i = 1, \overline{N-p}, \quad (2.39)$$

или в терминах элементов $\bar{h}_{ij}$ матрицы $\bar{H}$

$$f_i(t) = \sum_{j=1}^{p+1} \bar{h}_{ij} (x_{i+j-1} - t)_+^{p-1}/(p-1)!, \quad i = 1, \overline{N-p}.$$

Что из себя представляет функция $f_i(t)$? На каждом отрезке сетки это многочлен степени $p-1$. Далее, функция $f_i(t)$ имеет $p-2$ непрерывных производных, так как функции вида $(x_\kappa - t)_+^{p-1}$ принадлежит этому классу. Наконец, $f_i(t)$ — это локальная функция, сосредоточенная на интервале $[x_i, x_{i+p}]$. В самом деле, при $t > x_{i+p}$ все члены разложения (2.39) равны 0. При $t < x_i$ все выражения вида $(x_j - t)_+^{p-1}$ в (2.39) становятся обычными многочленами степени $p-1$. Вспомним, что h_{ij} выбираются из условия ортогональности таким многочленам.

Таким образом, матрица Грама A системы функций $f_1, f_2, \dots, f_{N-p}$ будет лепточкой с $2p-1$ ненулевыми диагоналями. Кроме того, она, конечно, симметрична и положительно определена. Для вычисления ее элементов a_{ij} необходимо вычислить интегралы

$$a_{ij} = \int_a^b f_i(t) f_j(t) dt$$

для $i \leq j \leq \kappa$, $\kappa = \min(i+p-1, N-p)$, $i = 1, 2, \dots, N-p$. Учитывая локальность функций $f_i(t)$, мы можем записать подробно:

$$a_{i,i+\kappa} = (f_i, f_{i+\kappa})_{L_2(a,b)} = \frac{1}{(p-1)!^2} \sum_{j=i+\kappa}^{i+p-1} \sum_{l=j-i+2}^{p+1} \sum_{m=j-i-\kappa+2}^{p+1} \times \\ \times \bar{h}_{il} \bar{h}_{i+\kappa,m} \int_{x_j}^{x_{j+1}} (x_{i+l-1} - t)^{p-1} (x_{i+\kappa+m-1} - t)^{p-1} dt, \quad (2.40)$$

где $i = \overline{1, N-p}$, $\kappa = \overline{0, \min(p-1, N-p-i)}$. В формуле (2.40) интегрирование производится только по тем ячей-

кам, где $f_i(t)$ и $f_{i+\kappa}(t)$ не равны 0 одновременно. При этом из разложения (2.39) удалены все члены, равные 0 в данной ячейке. Напомним, что в формуле (2.40) $\bar{h}_{il}$ — это элементы плотно упакованной матрицы $\bar{H}$, построенной в п. 2.3.1.

Вычисление интегралов, входящих в формулу (2.40), нерентабельно производить явно, раскрывая выражение в скобках под интегралом. Гораздо экономичнее употреблять для расчета квадратурную формулу Гаусса с p узлами, дающую точный результат на многочленах степени $2p - 1$. Это избавит от необходимости производить многочленные суммирования, нужно будет только подсчитать значения подынтегрального выражения в p точках (например, при $p = 5$ применение квадратурной формулы ускоряет счет в 20 раз). Точнее, замена переменных

$$t = (x_j + x_{j+1})/2 + \mu H/2, \quad H = x_{j+1} - x_j \quad (2.41)$$

переводит интеграл вида

$$\int_{x_j}^{x_{j+1}} (x_\kappa - t)^{p-1} (x_m - t)^{p-1} dt$$

в интеграл

$$(H/2)^{2p-1} \int_{-1}^1 [(A - \mu)(B - \mu)]^{p-1} d\mu, \quad (2.42)$$

где $A = (-x_j + 2x_\kappa - x_{j+1})/H$, $B = (-x_j + 2x_m - x_{j+1})/H$. Этот интеграл эффективно вычисляется по формуле Гаусса. Узлы и веса квадратурной формулы Гаусса целесообразно упаковать в некоторый массив, что и сделано в этой книге в п. 6.1.2 для $p = 1, 2, \dots, 10$. Можно пополнить эту таблицу, если не устраивает ограничение $p \leq 10$.

Упакуем теперь ненулевые элементы ленточной матрицы A в прямоугольный массив $\bar{A}[1:N-p, 0:p-1]$; помня о симметрии, храним только «половину» матрицы. Элементы a_{ij} матрицы A связаны с элементами $\bar{a}_{ij}$ массива $\bar{A}$ формулой

$$\bar{a}_{ij} = a_{i, i-j+p-1}. \quad (2.43)$$

Теперь несколько слов о структуре самого сплайна. Как мы знаем,

$$T\sigma = \frac{d^p \sigma}{dx^p} = \sum_{i=1}^{N-p} \lambda_i f_i \quad (2.44)$$

— кусочно-полиномиальная функция степени $p - 1$ гладкости $C^{p-2}(a, b)$. Из разложения (2.39) ясно, что

$$\sigma^{(p+\kappa)}(x_1) = \sigma^{(p+\kappa)}(x_N) = 0, \quad \kappa = 0, 1, \dots, p - 2. \quad (2.45)$$

За пределами отрезка $[x_1, x_N]$ функция $\sigma^{(p)}(t)$ тождественно нулевая. Следовательно, после p -кратного интегрирования мы получим сплайн-функцию $\sigma(t)$, которая в каждой ячейке вида (x_j, x_{j+1}) является многочленом степени $2p - 1$ и в интервалах (a, x_1) , (x_N, b) , если таковые есть, экстраполируется многочленом степени $p - 1$. На всем интервале (a, b) сплайн $\sigma(t)$ принадлежит классу гладкости C^{2p-2} .

2.3.4. Решение системы

После того, как сформированы матрицы A и H , необходимо решить систему

$$A\bar{\lambda} = H\bar{r} \quad (2.46)$$

в случае решения интерполяционной задачи или систему

$$(A + \alpha HH^*)\bar{\lambda}_\alpha = \bar{H}r \quad (2.47)$$

в случае решения задачи сглаживания. И в том, и в другом случае матрицы систем симметричные, положительно определенные и ленточные; в (2.46) матрица имеет ширину ленты $2p - 1$, в (2.47) — $2p + 1$.

Отметим сразу, что матрицы систем (2.46), (2.47) зависят только от сетки $x_1, x_2, \dots, x_N$ и не зависят от сеточных значений $r_1, r_2, \dots, r_N$. Поэтому решение систем предлагается производить в два этапа: сначала получить разложение Холецкого [14] для матрицы, затем решать серию задач с различными векторами входных данных r . Как мы увидим дальше, это позволит построить эффективные алгоритмы для решения задач сплайн-аппроксимации в многомерных областях типа параллелепипеда, где на линиях сетки приходится многократно производить одномерную аппроксимацию (см. § 2.6) и с ее помощью строить сплайн-функции многих переменных.

Для построения разложений Холецкого ленточных матриц существуют превосходные алгоритмы, входящие в комплект программ линейной алгебры, созданный Уилкинсоном и Райншем [14]. Можно, например, использовать пару программ: *CHOBANDDET* и *CHOBANDSOL*, первая из которых осуществляет разложение Холецкого

симметричной ленточной положительно определенной матрицы, вторая решает систему со многими правыми частями.

2.3.5. Вычисление полиномиального представления

После того, как решена система (2.46) или (2.47) относительно неизвестных $\lambda_1, \lambda_2, \dots, \lambda_{N-p}$, мы имеем явное представление

$$\frac{d^p \sigma}{dx^p} = \sum_{i=1}^{N-p} \lambda_i f_i(t) \quad (2.48)$$

или подробнее

$$\sigma^{(p)}(t) = \sum_{i=1}^{N-p} \sum_{l=1}^{p+1} \lambda_i \bar{h}_{il} (x_{i+l-1} - t)_+^{p-1} / (p-1)!. \quad (2.49)$$

С учетом локальности функций $f_i(t)$ можно записать на интервале $[x_j, x_{j+1}]$ функцию $\sigma^{(p)}(t) = \sigma_j^{(p)}(t)$ в виде

$$\sigma_j^{(p)}(t) = \sum_{i=\max(1, j-p+1)}^{\min(N-p, j)} \sum_{l=1}^{p+1} \lambda_i \bar{h}_{il} (x_{i+l-1} - t)_+^{p-1} / (p-1)!. \quad (2.50)$$

Наша цель состоит в получении кусочно-полиномиального представления сплайн-функции $\sigma(t)$. На интервале $[x_j, x_{j+1}]$ будем искать сплайн в виде

$$\sigma_j(t) = \sum_{\kappa=0}^{2p-1} A_j^{(\kappa)} (t - x_\kappa)^\kappa / \kappa!. \quad (2.51)$$

Ясно, что при этом коэффициенты $A_j^{(\kappa)}$ легко трактовать, именно

$$A_j^{(\kappa)} = \sigma^{(\kappa)}(x_j). \quad (2.52)$$

Из формулы (2.50) заключаем, что

$$\begin{aligned} A_j^{(p+\kappa)} = \sigma^{(p+\kappa)}(x_j) &= (-1)^\kappa \sum_{i=\max(1, j-p+1)}^{\min(N-p, j)} \sum_{l=1}^{p+1} \lambda_i \bar{h}_{il} \times \\ &\times (x_{i+l-1} - x_j)_+^{p-1-\kappa} / (p-1-\kappa)! \end{aligned} \quad (2.53)$$

при $j = 1, 2, \dots, N-1$, $\kappa = 0, 1, \dots, p-1$. Ясно, что суммирование в формуле (2.53) можно вести только при $j < i+l-1$. Таким образом, из (2.53) мы определяем ко-

коэффициенты многочлена

$$A_j^{(p)}, A_j^{(p+1)}, \dots, A_j^{(2p-1)}. \quad (2.54)$$

Кроме того, $A_j^{(0)} = r_j$ при интерполяции, в случае сглаживания $A_j^{(0)} = r_j^\alpha$. Величины r_j^α определяются из формулы (2.18). Осталось вычислить только коэффициенты

$$A_j^{(1)}, A_j^{(2)}, \dots, A_j^{(p-1)}. \quad (2.55)$$

Можно предложить много алгоритмов для их расчета, однако следует быть очень осторожным, так как большинство из них численно неустойчивы, причем, чем больше p , тем к более катастрофическим результатам они ведут. Практически к неустойчивому счету приводит любая попытка использовать уже посчитанный многочлен на соседнем интервале при движении по оси, подобно алгоритму решения задачи Коши. Главное требование — алгоритм расчета должен быть локальным, т. е. использовать только информацию с соседних интервалов, причем не переработанную, а исходную. В результате многочисленных экспериментов, сделанных автором (и не только им [8]), был выбран алгоритм, основанный на следующем обстоятельстве.

Пусть $P_j(t)$ — многочлен степени $2p - 1$, совпадающий на отрезке $[x_j, x_{j+1}]$ со сплайн-функцией $\sigma(t)$. Ясно, что $\sigma(t)$ может быть записана в виде

$$\sigma(t) = P_{p-1}(t) + \sum_{i=1}^{N-p} \sum_{l=1}^{p+1} \lambda_i \bar{h}_{il} (x_{i+l-1} - t)_+^{2p-1} / (2p-1)!, \quad (2.56)$$

где $P_{p-1}(t)$ — некоторый многочлен степени $p - 1$. Эта формула является следствием p -кратного интегрирования формулы (2.49). Тогда

$$P_j(t) = \sum_{i+l-1 > j} \lambda_i \bar{h}_{il} (x_{i+l-1} - t)^{2p-1} / (2p-1)! + P_{p-1}(t). \quad (2.57)$$

Следовательно,

$$\begin{aligned} P_{j+1}(t) - P_j(t) &= \sum_{i+l-1 > j+1} - \sum_{i+l-1 > j} = \left(- \sum_{i+l-1=j} \lambda_i \bar{h}_{il} \right) \times \\ &\times (x_{j+1} - t)^{2p-1} / (2p-1)! = c_j (x_{j+1} - t)^{2p-1} / (2p-1)!. \end{aligned} \quad (2.58)$$

С другой стороны, мы знаем, что

$$\begin{aligned} P_j^{(2p-1)}(t) &= A_j^{(2p-1)} = \text{const}, \\ P_{j+1}^{(2p-1)}(t) &= A_{j+1}^{(2p-1)} = \text{const}, \end{aligned}$$

поэтому неизбежно

$$c_j = A_{j+1}^{(2p-1)} - A_j^{(2p-1)}.$$

Таким образом, многочлены P_j и P_{j+1} отличаются на полином, который нам уже известен. Рассмотрев $(p-1)$ принадлежащих к x_j точек сетки, например $x_{j+1}, x_{j+2}, \dots, x_{j+p-1}$, можно определить недостающие коэффициенты (2.55), исходя из условий

$$P_m(x_m) = r_m, \quad m = j+1, j+2, \dots, j+p-1. \quad (2.59)$$

Условия (2.59) с учетом формул (2.58) приводят к линейной алгебраической системе порядка $p-1$ относительно коэффициентов $A_j^{(1)}, A_j^{(2)}, \dots, A_j^{(p-1)}$. В самом деле, из (2.51) получаем

$$\begin{aligned} \sum_{\kappa=1}^{p-1} A_j^{(\kappa)} (x_m - x_j)^\kappa / \kappa! &= r_m - \sum_{\kappa=p}^{2p-1} A_j^{(\kappa)} (x_m - x_j)^\kappa / \kappa! - A_j^{(0)} - \\ &- \sum_{n=2}^{p-1} (A_{j+n}^{(2p-1)} - A_{j+n-1}^{(2p-1)}) (x_{j+1} - x_{j+n})^{2p-1} / (2p-1)!, \end{aligned} \quad (2.60)$$

при этом $m = j+1, j+2, \dots, j+p-1$. На практике p невелико ($p = 2, 3, 4$); система (2.60) может быть успешно решена с помощью стандартной процедуры обеспечения ЭВМ, на которой Вы работаете. При этом желательно опять использовать процедуры, позволяющие решать системы со многими правыми частями. Это обеспечит одновременное решение серии задач.

Следует отметить, что в процессе эксплуатации программ, использующих такой алгоритм расчета полиномиального представления сплайна, ни разу не был зарегистрирован эффект численной неустойчивости.

§ 2.4. СПЛАЙНЫ ЧЕТНОЙ СТЕПЕНИ НА ОТРЕЗКЕ И АППРОКСИМАЦИЯ ПО ЛОКАЛЬНЫМ ИНТЕГРАЛАМ

Аппроксимация функций по средним интегральным значениям обычно используется в том случае, когда экспериментальная кривая сильно зашумлена и измеренные

значения в узлах недостоверны. В таком случае вместо значений функции в узлах сетки в качестве исходной информации рассматриваются значения интегралов от экспериментальной кривой по некоторой системе интервалов. При интегрировании высокочастотные экспериментальные ошибки взаимно аннулируются, и к средним интегральным значениям можно относиться с большим доверием, нежели к значениям функции. Поставим задачу формально.

Рассмотрим отрезок $[a, b]$, $-\infty < a < b < \infty$, с нанесенной на него сеткой $x_1, x_2, \dots, x_N$,

$$a \leq x_1 < x_2 < \dots < x_N \leq b. \quad (2.61)$$

Положим $X = W_2^p[a, b]$, $p > 0$, $Y = L_2[a, b]$, $T = \frac{d^p}{dx^p}$. Пусть нам заданы интегральные средние $r_1, r_2, \dots, r_{N-1}$ на разбиении (2.61), т. е. функционалы k_i таковы:

$$(k_i, u)_{W_2^p} = \frac{1}{x_{i+1} - x_i} \int_{x_i}^{x_{i+1}} u(x) dx, \quad i = \overline{1, N-1}. \quad (2.62)$$

Поставим задачу отыскания интерполяционной сплайн-функции $\sigma(x) \in W^p[a, b]$ из условий

$$\begin{aligned} \frac{1}{x_{i+1} - x_i} \int_{x_i}^{x_{i+1}} \sigma(x) dx &= r_i, \quad i = \overline{1, N-1}, \\ \int_a^b [\sigma^{(p)}(x)]^2 dx &= \min \end{aligned} \quad (2.63)$$

или задачу отыскания сглаживающей сплайн-функции $\sigma_\alpha(x) \in W_2^p[a, b]$ с параметром сглаживания $\alpha > 0$ из условия

$$\alpha \int_a^b [\sigma_\alpha^{(p)}(x)]^2 dx + \sum_{i=1}^{N-1} \left[\frac{1}{x_{i+1} - x_i} \int_{x_i}^{x_{i+1}} \sigma_\alpha(x) dx - r_i \right]^2 = \min. \quad (2.64)$$

Как мы уже знаем (см. § 1.3), задачи (2.63) и (2.64) имеют единственное решение при $N \geq p + 1$. Построим решения этих задач в соответствии с алгоритмами § 2.1, 2.2.

2.4.1. Формирование матрицы H

Ядро оператора $T = \frac{dp}{dx^p}$ — все многочлены степени не выше $p - 1$, размерность ядра q равна p , базис ядра составляют функции $1, x, \dots, x^{p-1}$. Будем искать функций $h_1, h_2, \dots, h_{N-p-1}$ в виде

$$h_i = \sum_{j=i}^{i+p} h_{ij} k_j. \quad (2.65)$$

Коэффициенты h_{ij} следует выбрать из условия ортогональности:

$$(h_i, x^\kappa)_{W_2^p} = 0, \quad \kappa = 0, 1, \dots, p - 1, \quad (2.66)$$

или подробно

$$\sum_{j=i}^{i+p} \frac{h_{ij}}{x_{j+1} - x_j} \int_{x_j}^{x_{j+1}} x^\kappa dx = 0, \quad \kappa = 0, 1, \dots, p - 1, \quad (2.67)$$

т. е.

$$\sum_{j=i}^{i+p} \frac{h_{ij}}{x_{j+1} - x_i} (x_{j+1}^{\kappa+1} - x_j^{\kappa+1}) = 0, \quad \kappa = 0, 1, \dots, p - 1. \quad (2.68)$$

Запишем формулу (2.68) иначе:

$$\begin{aligned} - \frac{h_{ii}}{x_{i+1} - x_i} x_i^{\kappa+1} + \sum_{j=i+1}^{i+p} \left(\frac{h_{ij-1}}{x_j - x_{j-1}} - \frac{h_{ij}}{x_{j+1} - x_j} \right) x_j^{\kappa+1} + \\ + \frac{h_{i+p}}{x_{i+p+1} - x_{i+p}} x_{i+p+1}^{\kappa+1} = 0. \end{aligned} \quad (2.69)$$

Построим теперь коэффициенты $\tilde{h}_{ij}$, удовлетворяющие условию

$$\sum_{j=i}^{i+p+1} \tilde{h}_{ij} x_j^\kappa = 0, \quad \kappa = 0, 1, \dots, p.$$

Коэффициенты $\tilde{h}_{ij}$ могут быть выбраны как коэффициенты разностного аналога $(p + 1)$ -й производной на сеточном шаблоне из $p + 2$ точек $x_i, x_{i+1}, \dots, x_{i+p+1}$, причем разностная схема должна иметь порядок аппроксимации $O(h^{p+1})$. Коэффициенты генерируются с помощью разделенных разностей, как было описано в п. 2.3.1. Зная $\{\tilde{h}_{ij}\}_{j=i}^{i+p+1}$, мы можем вычислить $\{h_{ij}\}_{j=i}^{i+p}$ по формулам, непо-

средственно следующим из (2.69) (напоминаем, что h_{ij} вычисляются с точностью до постоянного множителя):

$$\begin{aligned} h_{ii} &= -\tilde{h}_{ii}, \\ h_{ij} &= -\tilde{h}_{ij} - h_{i, j-1}, \quad j = i + 1, \dots, i + p. \end{aligned} \quad (2.70)$$

Если перенумеровать коэффициенты h_{ij} по формуле

$$\bar{h}_{ij} = h_{i, i+j-1}, \quad i = 1, 2, \dots, N - p - 1; \quad j = 1, 2, \dots, p + 1,$$

то они могут быть упакованы в прямоугольную матрицу $\bar{H}$ размерности $(N - p - 1) \times (p + 1)$. Она содержит в себе «ленту» из ненулевых элементов матрицы H .

2.4.2. Формирование матрицы A

Теперь необходимо вычислить функции $f_i(t)$, $i = 1, 2, \dots, N - p - 1$, по формуле

$$f_i(t) = \left(\frac{d^p}{dx^p} \right)^{* - 1} h_i(t). \quad (2.71)$$

На основании общей формулы (2.30), выведенной в п. 2.3.2, имеем

$$f_i(t) = (h_i(x), (x - t)_+^{p-1} / (p - 1)!)_{W_2^p} \quad (2.72)$$

или подробно

$$f_i(t) = \frac{1}{p!} \sum_{j=1}^{p+1} \frac{\bar{h}_{ij}}{x_{i+j} - x_{i+j-1}} ((x_{i+j} - t)_+^p - (x_{i+j-1} - t)_+^p). \quad (2.73)$$

$f_i(t)$ — функция с локальным носителем, сосредоточенным в отрезке $[x_i, x_{i+p+1}]$, класса гладкости C^{p-1} , составленная из полиномов степени p на каждом отрезке сетки. Матрица Грама системы функций $f_1, f_2, \dots, f_{N-p-1}$ будет симметричной положительно определенной ленточной матрицей с $2p + 1$ ненулевыми диагоналями. Для определения ее элементов следует вычислить интегралы

$$a_{ij} = \int_a^b f_i(t) f_j(t) dt$$

для $i \leq j \leq \min(i + p, N - p - 1)$. С учетом локальности функций

$$a_{i,i+\kappa} = \frac{1}{(p!)^2} \sum_{j=i+\kappa}^{i+p} \sum_{l=j+1}^{i+p+1} \sum_{m=j+1}^{i+\kappa+p+1} \left(\frac{\bar{h}_{il-i}}{x_l - x_{l-1}} - \frac{\bar{h}_{il-i+1}^0}{x_{l+1} - x_l} \right) \times \\ \times \left(\frac{\bar{h}_{i+\kappa,m-i+\kappa}}{x_m - x_{m-1}} - \frac{\bar{h}_{i+\kappa,m-i-\kappa+1}^0}{x_{m+1} - x_m} \right) \times \int_{x_j}^{x_{j+1}} (x_l - t)^p (x_m - t)^p dt, \quad (2.74)$$

где $i = 1, 2, \dots, N - p - 1$, $\kappa = 0, \overline{\min(p, N - p - 1 - i)}$.
Здесь

$$\bar{h}_{s,c}^0 = \begin{cases} 0, & c = p + 2, \\ \bar{h}_{s,c}, & c \neq p + 2. \end{cases}$$

В формуле (2.74) интегрирование ведется только по интервалам, где $f_i(t)$ и $f_{i+\kappa}(t)$ не равны 0 одновременно.

Так же, как и в § 2.3, скоростной расчет интегралов следует вести с использованием квадратных формул Гаусса с p узлами.

Ненулевые элементы ленточной матрицы A с учетом симметрии могут быть упакованы в прямоугольный массив $\bar{A}[1 : N - p - 1, 0 : p]$, элементы $\bar{a}_{ij}$ которого связаны с a_{ij} формулой

$$\bar{a}_{ij} = a_{i, i-j+p}. \quad (2.75)$$

2.4.3. Решение системы и вычисление полиномиального представления

Как обычно, при решении в случае интерполяции системы

$$\bar{A}\bar{\lambda} = H\bar{r}$$

или в случае сглаживания системы

$$(A + \alpha HH^*)\bar{\lambda}_\alpha = H\bar{r}$$

с симметричными положительно определенными ленточными матрицами рентабельно применять прямые методы, которые сначала дают разложение Холецкого матриц A или $A + \alpha HH^*$, что позволяет в дальнейшем решать серию систем с различными правыми частями. По-прежнему весьма удобен комплект программ Уилкинсона, Райнша [14].

После того, как система решена, отыскивается полиномиальное представление сплайна в виде

$$\sigma_m(t) = \sum_{j=0}^{2p} A_m^{(j)} (t - x_m)^j / j! \quad (2.76)$$

на каждом отрезке $[x_m, x_{m+1}]$. Ясно, что

$$A_m^{(j)} = \sigma^{(j)}(x_m). \quad (2.77)$$

Величина $\sigma^{(p)}(t)$ может быть записана в виде

$$\begin{aligned} \sigma_m^{(p)}(t) = & \sum_{i=\max(1, m-p)}^{\min(N-p-1, m)} \sum_{l=1}^{p+1} \lambda_i \frac{\bar{h}_{il}}{x_{i+l} - x_{i+l-1}} \times \\ & \times \frac{(x_{i+l} - t)_+^p - (x_{i+l-1} - t)_+^p}{p!}. \end{aligned} \quad (2.78)$$

Отсюда немедленно получаем

$$\begin{aligned} A_m^{(p+j)} = \sigma_m^{(p+j)}(x_m) = & (-1)^j \sum_{i=\max(1, m-p)}^{\min(N-p-1, m)} \sum_{l=1}^{p+1} \times \\ & \times \lambda_i \frac{\bar{h}_{il}}{x_{i+l} - x_{i+l-1}} \frac{(x_{i+l} - x_m)_+^{p-j} - (x_{i+l-1} - x_m)_+^{p-j}}{(p-j)!} \end{aligned} \quad (2.79)$$

при $m = 1, 2, \dots, N-1$; $j = 0, 1, \dots, p$ (суммирование ведется только при $m < i+l$). Таким образом, коэффициенты

$$A_m^{(p)}, A_m^{(p+1)}, \dots, A_m^{(2p)}$$

теперь известны, осталось определить

$$A_m^{(0)}, A_m^{(1)}, \dots, A_m^{(p-1)}. \quad (2.80)$$

Рассуждая как в п. 2.3.5, получим, что полиномиальные представления сплайна $P_m(t)$ и $P_{m+1}(t)$ на отрезках $[x_m, x_{m+1}]$ и $[x_{m+1}, x_{m+2}]$ связаны условием

$$P_{m+1}(t) - P_m(t) = c_m (t - x_{m+1})^{2p} / (2p)!,$$

где $c_m = A_{m+1}^{(2p)} - A_m^{(2p)}$. Таким образом, рассмотрев $p-1$ интервалов, прилегающих к $[x_m, x_{m+1}]$ справа или слева, можно определить недостающие коэффициенты (2.80), исходя из условий

$$\frac{1}{x_{\kappa+1} - x_{\kappa}} \int_{x_{\kappa}}^{x_{\kappa+1}} P_{\kappa}(x) dx = \begin{cases} r_{\kappa}, & \alpha = 0; \\ r_{\kappa}^{\alpha}, & \alpha > 0. \end{cases} \quad (2.81)$$

Они приводят к линейной алгебраической системе поряд-

ка p относительно переменных (2.80). Так как p обычно невелико, можно использовать стандартные процедуры ЭВМ, на которой Вы работаете. Желательно использовать алгоритмы, позволяющие решать системы со многими правыми частями.

§ 2.5. СПЛАЙН-ИНТЕРПОЛЯЦИЯ В ОБЛАСТЯХ ТИПА ПАРАЛЛЕЛЕПИПЕДА

В § 2.3 мы рассмотрели эффективный алгоритм построения интерполяционного сплайна нечетной степени на отрезке. Алгоритм был ориентирован на решение серии таких задач. В этом параграфе мы проиллюстрируем технику построения интерполяционного сплайна двух переменных, которая очевидным образом переносится на сплайны многих переменных.*)

Пусть $\Omega = [a, b] \times [c, d]$ — прямоугольник на плоскости; введем в рассмотрение две сетки

$$a = x_1 < x_2 < \dots < x_N = b, \quad c = y_1 < y_2 < \dots < y_M = d \quad (2.82)$$

по переменным x и y соответственно. Пусть в узлах сетки (x_i, y_j) заданы значения f_{ij} сеточной функции. Поставим задачу отыскания кусочно-полиномиальной функции $\sigma(x, y)$, которая в каждой ячейке сетки $\Omega_{ij} = [x_i, x_{i+1}] \times [y_j, y_{j+1}]$ является полиномом степени $2p-1$ по x и $2q-1$ по y , т. е.

$$\sigma|_{\Omega_{ij}} = \sigma_{ij}(x, y) = \sum_{k=0}^{2p-1} \sum_{l=0}^{2q-1} a_{kl}^{ij} \frac{(x-x_i)^k (y-y_j)^l}{k!l!}. \quad (2.83)$$

При этом $\sigma(x, y)$ имеет непрерывные производные по x до порядка $2p-2$ и по y до порядка $2q-2$ включительно. Сплайн-функция удовлетворяет следующим краевым условиям:

$$\left. \frac{\partial^r \sigma}{\partial x^r} \right|_{x=a} = \left. \frac{\partial^r \sigma}{\partial x^r} \right|_{x=b} = 0, \quad r = p, p+1, \dots, 2p-2; \quad (2.84)$$

$$\left. \frac{\partial^k \sigma}{\partial y^k} \right|_{y=c} = \left. \frac{\partial^k \sigma}{\partial y^k} \right|_{y=d} = 0, \quad k = q, q+1, \dots, 2q-2. \quad (2.85)$$

Опишем алгоритм построения такой функции $\sigma(x, y)$ при условии, что в нашем распоряжении уже имеется метод

*) Вариационная постановка таких задач приведена в [33].

построения сплайна нечетной степени на отрезке (см. § 2.3).

При фиксированном x^* функция $\sigma(x^*, y)$ является сплайном степени $2q - 1$ по y с краевыми условиями

$$\frac{d^\kappa \sigma(x^*, y)}{dy^\kappa} = 0, \quad \kappa = q, q + 1, \dots, 2q - 2. \quad (2.86)$$

Аналогично при фиксированном y^* функция $\sigma(x, y^*)$ — сплайн степени $2p - 1$ по x с краевыми условиями

$$\frac{d^\kappa \sigma(x, y^*)}{dx^\kappa} = 0, \quad \kappa = p, p + 1, \dots, 2p - 2. \quad (2.87)$$

Воспользуемся этим обстоятельством, а также тем, что в искомом разложении (2.83) коэффициенты $a_{\kappa l}^{ij}$ удовлетворяют очевидному условию

$$a_{\kappa l}^{ij} = \frac{\partial^{\kappa+l} \sigma_{ij}}{\partial x^\kappa \partial y^l}(x_i, y_j). \quad (2.88)$$

На линиях сетки $x = x_i$ решим N одномерных задач ($i = 1, 2, \dots, N$) построения сплайнов $g_i(y)$ степени $2q - 1$ таких, что

$$g_i(y_\kappa) = f_{i\kappa}, \quad \kappa = 1, 2, \dots, M. \quad (2.89)$$

Все эти задачи решаются на одной и той же сетке

$$c = y_1 < y_2 < \dots < y_M = d, \quad (2.90)$$

поэтому только один раз выполняется LU -разложение соответствующей матрицы A (см. § 2.3) после чего сплайны $g_i(y)$ находятся одновременно. Если в интервале $[y_j, y_{j+1}]$

$$g_i|_{[y_j, y_{j+1}]} = g_{ij}(y) = \sum_{\kappa=0}^{2q-1} h_\kappa (y - y_j)^\kappa / \kappa!, \quad (2.91)$$

то, очевидно,

$$a_{0\kappa}^{ij} = \frac{\partial^\kappa g_{ij}}{\partial y^\kappa}(y_j) = h_\kappa, \quad \kappa = 0, 1, \dots, 2q - 1. \quad (2.92)$$

Таким образом, после решения N задач сплайн-интерполяции на линиях $x = x_i$ нам станут известны частные производные по y до порядка $2q - 1$ включительно в узлах сетки.

Искомая функция $\sigma(x, y)$ после ее дифференцирования по y ($2q - 1$) раз все равно остается кусочно-полино-

миальным сплайном степени $2p - 1$ по переменной x . Если теперь на M линиях сетки $y = y_j$ произвести интерполирование значений $f_{ij} = a_{00}^{ij}$, а также значений производных a_{0k}^{ij} , $k = 1, 2, \dots, 2q - 1$, то в результате мы получим значения всех производных $\frac{\partial^{k+l} \sigma_{ij}}{\partial x^k \partial y^l}$, $l = 0, 1, \dots, 2p - 1$, в узлах сетки, т. е. коэффициенты a_{kl}^{ij} , по которым в соответствии с формулой (2.83) во всех ячейках сетки восстанавливается сплайн-функция $\sigma(x, y)$. Опять-таки отметим, что при решении $2qM$ одномерных задач интерполяции по переменной x необходимо только один раз выполнить LU -разложение матрицы A , связанной с сеткой $a = x_1 < x_2 < \dots < x_N = b$, после чего задачи решаются одновременно.

Итак, при реализации алгоритма нужно решить N одномерных задач на сетке из M узлов и $2qM$ одномерных задач на сетке из N узлов, причем сначала строятся сплайны степени $2q - 1$, а затем степени $2p - 1$.

Разумеется, алгоритм можно было провести в другом порядке: сначала строить сплайны на линиях $y = y_j$, а потом на линиях $x = x_i$. Результат был бы тем же. Возникает вопрос: при каких соотношениях N и M , p и q следует выбрать тот или иной способ. Анализ количества арифметических операций показывает, что при $p \ll N$, $q \ll M$ это безразлично. Однако в реальной ситуации, когда $p, q = 2, 3$; N, M порядка 10, 20, все-таки предпочтительнее сначала провести аппроксимацию по линиям с большим количеством узлов, а затем решать одновременные задачи на сетке с меньшим количеством узлов. Если p и q отличаются значительно, то следует сначала решать задачи построения сплайнов более высокой, а уж потом — более низкой степени.

Описанный алгоритм построения сплайна в прямоугольнике может быть обобщен на произвольный n -мерный параллелепипед. Если на нем нанесена параллелепипедальная сетка, содержащая N_i узлов по i -му направлению, то в каждой ячейке сплайн будет записан в виде

$$\sigma(X) = \sum_{\alpha=0}^{2P-1} a_{\alpha}^{\beta} \frac{(X - X_{\beta})^{\alpha}}{\alpha!}, \quad (2.93)$$

где мультииндексы α, β , P , $0, 1$, переменная X и точка сетки X_{β} имеют вид:

$$\begin{aligned}
X &= (x^{(1)}, x^{(2)}, \dots, x^{(n)}), \\
X_\beta &= (x_{\beta_1}^{(1)}, x_{\beta_2}^{(2)}, \dots, x_{\beta_n}^{(n)}), \quad \beta_i = 1, 2, \dots, N_i, \\
\beta &= (\beta_1, \beta_2, \dots, \beta_n), \\
\alpha &= (\alpha_1, \alpha_2, \dots, \alpha_n), \\
P &= (p_1, p_2, \dots, p_n), \\
0 &= (0, 0, \dots, 0), \\
1 &= (1, 1, \dots, 1).
\end{aligned} \tag{2.94}$$

Аналогично двумерному случаю коэффициенты

$$a_\alpha^\beta = \frac{\partial^\alpha \sigma}{\partial X^\alpha} (X_\beta) \tag{2.95}$$

могут быть найдены путем решения серий одномерных задач интерполирования на линиях сетки самой функции, а также ее производных.

§ 2.6. ФУНКЦИИ ГРИНА И АНАЛИТИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ СПЛАЙНОВ В МНОГОМЕРНЫХ ОБЛАСТЯХ ПРОИЗВОЛЬНОЙ КОНФИГУРАЦИИ

Пусть Ω — некоторая ограниченная n -мерная область, $X(\Omega)$, $Y(\Omega)$ — два функциональных гильбертовых пространства, $T: X(\Omega) \rightarrow Y(\Omega)$ — линейный ограниченный оператор. Определим на $X(\Omega)$ линейные независимые ограниченные функционалы $k_1, k_2, \dots, k_N$. Будем предполагать, что для любых чисел $r_1, r_2, \dots, r_N$ задача сплайн-интерполяции

$$\begin{aligned}
(k_i, \sigma)_{X(\Omega)} &= r_i, \quad i = \overline{1, N}, \\
\|T\sigma\|_{Y(\Omega)}^2 &= \min
\end{aligned} \tag{2.96}$$

имеет единственное решение $\sigma \in X(\Omega)$ (см. § 1.2).

Рассмотрим оператор $T^*T: X(\Omega) \rightarrow X(\Omega)$. Пусть существует функция Грина $G(x, t)$ этого оператора, обладающая свойством

$$T^*TG(x, t) = \delta_t, \tag{2.97}$$

где $\delta_t \in X(\Omega)$ такова, что для любой $v \in X(\Omega)$, $t \in \Omega$, $(\delta_t, v)_X = v(t)$. Здесь δ_t — функция Дирака, сосредоточенная в точке t . Отметим, что в этом случае функция $G(x, t)$ всегда симметрична, $G(x, t) = G(t, x)$.

Теорема 2.2. Сплайн-функция $\sigma(P)$, являющаяся решением задачи (2.96), представима в виде

$$\sigma(P) = \sum_{i=1}^N \lambda_i (k_i(x), G(x, P))_{X(\Omega)} + n(P), \quad (2.98)$$

где $n(P)$ — элемент ядра оператора T , λ_i — некоторые вещественные коэффициенты.

Доказательство. Запишем условие ортогональности для интерполяционного сплайна (см. § 1.1)

$$(T\sigma, Ty)_{Y(\Omega)} = 0, \quad \forall y \in N(A),$$

где $A: X \rightarrow E_N$ оператор, действующий по формуле

$$Au = [(k_1, u)_X, (k_2, u)_X, \dots, (k_N, u)_X].$$

Условие ортогональности является необходимым и достаточным для того, чтобы элемент $\sigma \in X(\Omega)$ был интерполяционным сплайном с каким-нибудь вектором входных данных $[r_1, r_2, \dots, r_N]$. Подставим разложение (2.98) в условие ортогональности

$$\begin{aligned} \forall y \in N(A) \quad (T\sigma, Ty)_{Y(\Omega)} &= (T^*T\sigma, y)_{X(\Omega)} = \\ &= \sum_{i=1}^N \lambda_i (T^*T(k_i(x), G(x, P))_{X(\Omega)}, y)_{X(\Omega)} = \\ &= \sum_{i=1}^N \lambda_i ((k_i(x), \delta_P(x))_{X(\Omega)}, y)_{X(\Omega)} = \sum_{i=1}^N \lambda_i (k_i(P), y)_{X(\Omega)} = 0, \end{aligned}$$

что и требовалось доказать.

Предположим теперь, что ядро $N(T)$ оператора T конечномерно, базис ядра составляют функции $n_1(P)$, $n_2(P)$, $\dots$, $n_q(P)$. Сплайн представим в виде

$$\sigma(P) = \sum_{i=1}^N \lambda_i g_i(P) + \sum_{\kappa=1}^q \nu_\kappa n_\kappa(P), \quad (2.99)$$

где $g_i(P) = (k_i(x), G(x, P))_{X(\Omega)}$. Очевидно, что

$$(T^*T\sigma)(P) = \sum_{i=1}^N \lambda_i (k_i(x), T^*TG(x, P))_X = \sum_{i=1}^N \lambda_i k_i(P). \quad (2.100)$$

Умножая это равенство скалярно на $n_1, n_2, \dots, n_q$, получим

$$(T^*T\sigma, n_\kappa)_X = (T\sigma, Tn_\kappa)_Y = 0 = \sum_{i=1}^N \lambda_i (k_i, n_\kappa)_X. \quad (2.101)$$

Привлекая интерполяционное условие $(\sigma, k_i)_X = r_i$, $i = 1$,

2, ..., N, и формулу (2.101), получим следующую линейную алгебраическую систему порядка $(N + q)$ для определения коэффициентов λ_i, v_k :

$$\begin{vmatrix} \bar{A} & \bar{B} \\ \bar{B}^* & 0 \end{vmatrix} \begin{vmatrix} \bar{\lambda} \\ \bar{v} \end{vmatrix} = \begin{vmatrix} \bar{r} \\ 0 \end{vmatrix}, \quad (2.102)$$

где

$$\bar{\lambda} = \begin{vmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{vmatrix}, \quad \bar{v} = \begin{vmatrix} v_1 \\ v_2 \\ \vdots \\ v_q \end{vmatrix}, \quad \bar{r} = \begin{vmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \end{vmatrix},$$

$$\bar{A} = \begin{vmatrix} (k_1, g_1)_X & (k_1, g_2)_X & \dots & (k_1, g_N)_X \\ (k_2, g_1)_X & (k_2, g_2)_X & \dots & (k_2, g_N)_X \\ \vdots & \vdots & \ddots & \vdots \\ (k_N, g_1)_X & (k_N, g_2)_X & \dots & (k_N, g_N)_X \end{vmatrix}, \quad \bar{B} = \begin{vmatrix} (k_1, n_1)_X & \dots & (k_1, n_q)_X \\ (k_2, n_1)_X & \dots & (k_2, n_q)_X \\ \vdots & \ddots & \vdots \\ (k_N, n_1)_X & \dots & (k_N, n_q)_X \end{vmatrix}$$

Система (2.102) всегда имеет симметрическую матрицу, т. е. $(k_i, g_j)_X = (k_j, g_i)_X$. В самом деле,

$$(k_i, g_j)_X = (T^* T g_i, g_j)_X = (T g_i, T g_j)_Y. \quad (2.103)$$

Симметрия последнего выражения не вызывает сомнений. Для отыскания интерполяционного сплайна σ необходимо решить систему (2.102) с симметрической матрицей. Разумеется, матрица невырождена, так как исходная интерполяционная задача имеет единственное решение.

Выведем теперь систему линейных алгебраических уравнений для определения сглаживающего сплайна σ_α , являющегося решением задачи минимизации функционала

$$\alpha \|Tu\|_{Y_1}^2 + \sum_{i=1}^N [(k_i, u)_X - r_i]^2$$

на пространстве $X(\Omega)$. Ясно, что сглаживающий сплайн, являющийся некоторым интерполяционным (см. § 1.2), также представим в виде (2.99). Для сглаживающего сплайна σ_α также справедливы равенства (2.100), (2.101), т. е. в матричной форме

$$\bar{B}^* \bar{\lambda} = 0. \quad (2.104)$$

Введем в рассмотрение оператор $A^* : E_N \rightarrow X$, сопряжен-

ный по отношению к $A : X \rightarrow E_N$. Так как

$$(Ax, y)_{E_N} = \sum_{i=1}^N y_i (k_i, x)_X = \left(\sum_{i=1}^N y_i k_i, x \right)_X = (x, A^*y)_X,$$

оператор A^* действует по формуле

$$A^*y = \sum_{i=1}^N y_i k_i \quad (2.105)$$

и имеет нулевое ядро в силу линейной независимости функционалов k_i . Равенство (2.100) можно переписать в виде

$$T^*T\sigma = A^*\bar{\lambda}. \quad (2.106)$$

Запишем условие ортогональности (см. § 1.2) для сглаживающих сплайнов:

$$\forall x \in X \quad \alpha(T\sigma_\alpha, Tx)_Y + (A\sigma_\alpha - \bar{r}, Ax) = 0$$

или

$$(\alpha T^*T\sigma_\alpha + A^*(A\sigma_\alpha - \bar{r}), x) = 0. \quad (2.107)$$

С учетом (2.106)

$$\alpha A^*\bar{\lambda} + A^*(A\sigma_\alpha - \bar{r}) = 0. \quad (2.108)$$

Оператор A^* имеет нулевое ядро, следовательно,

$$\alpha\bar{\lambda} + A\sigma_\alpha = \bar{r}. \quad (2.109)$$

Условие (2.109) может быть переписано в матричной форме:

$$(\alpha I + \bar{A})\bar{\lambda} + \bar{B}\bar{v} = \bar{r}.$$

Совместно с уравнениями (2.104) получаем итоговую систему

$$\begin{vmatrix} \alpha I + \bar{A} & \bar{B} \\ \bar{B}^* & 0 \end{vmatrix} \begin{vmatrix} \bar{\lambda} \\ \bar{v} \end{vmatrix} = \begin{vmatrix} \bar{r} \\ 0 \end{vmatrix}. \quad (2.110)$$

Здесь I — единичная матрица размерности N . Матрица системы (2.110) снова симметрична и невырождена.

§ 2.7. СПЛАЙН-АППРОКСИМАЦИЯ В ОБЛАСТЯХ С ХАОТИЧЕСКИ РАСПОЛОЖЕННЫМИ ИНТЕРПОЛЯЦИОННЫМИ УЗЛАМИ

Пусть Ω — n -мерная ограниченная область, $P_1, P_2, \dots, P_N$ — система расположенных в Ω точек, в которых нам известны значения $r_1, r_2, \dots, r_N$. Введем в рассмотре-

ние пространство $X = W_2^m(\Omega)$, причем $m > n/2$. Это неравенство обеспечивает вложение $W_2(\Omega)$ в $C(\Omega)$ [9], или, что то же самое, непрерывность функционалов

$$(k_i, u)_{W_2^m} = u(P_i), \quad (2.111)$$

дающих значение функции $u \in W_2^m(\Omega)$ в любой точке $P \in \Omega$. Поставим задачу отыскания интерполяционного сплайна $\sigma(P) \in W_2^m(\Omega)$ из условий

$$\sigma(P_i) = r_i, \quad i = \overline{1, N};$$

$$\int_{\Omega} \sum_{|\alpha|=m} \frac{m!}{\alpha!} (D^\alpha \sigma)^2 d\Omega = \min_{W_2^m}, \quad (2.112)$$

где $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ — мультииндекс. В § 1.3 мы уже проанализировали вопросы существования и единственности решения этой задачи. Найдем теперь аналитическое представление ее решения. Ясно, что в данном случае $T^*T = (-1)^m \Delta^m$, где Δ — оператор Лапласа.*) Функция Грина $G_{m,n}$ такого оператора хорошо известна и с точностью до постоянного множителя, который совершенно не важен в будущем разложении сплайна, имеет вид

$$G_{m,n}(x, P) = G_{m,n}(x - P) = \begin{cases} \|x - P\|^{2m-n} \ln \|x - P\|, & n \text{ четное} \\ \|x - P\|^{2m-n} \cdot n & n \text{ нечетное.} \end{cases} \quad (2.113)$$

Здесь

$$\|x - P\| = \left[\sum_{i=1}^n (x^{(i)} - P^{(i)})^2 \right]^{1/2},$$

$$x = (x^{(1)}, x^{(2)}, \dots, x^{(n)}), \quad P = (P^{(1)}, P^{(2)}, \dots, P^{(n)}).$$

С учетом результатов § 2.7 сплайн представим в виде

$$\sigma(P) = \sum_{i=1}^n \lambda_i G_{m,n}(P - P_i) + \sum_{|\alpha| \leq m-1} v_\alpha P^\alpha, \quad (2.114)$$

где $P^\alpha = ((P^{(1)})^{\alpha_1} \cdot (P^{(2)})^{\alpha_2} \cdot \dots \cdot (P^{(n)})^{\alpha_n})$. Выпишем систему линейных алгебраических уравнений для плоской об-

*) Совпадение операторов понимается в смысле совпадения билинейных форм $(T^*Tu, \varphi)_X = (u, (-1)^m \Delta^m \varphi)_{L_2}$, где $u \in W_2^m$, φ — бесконечно дифференцируемая финитная в Ω функция.

ласти ($n = 2$) при $m = 2$, что соответствует условию

$$\int_{\Omega} [\sigma_{xx}^2 + 2\sigma_{xy}^2 + \sigma_{yy}^2] d\Omega = \min_{w_2^2}.$$

Если точка P_i имеет координаты (x_i, y_i) , то система приобретает вид

$$\begin{vmatrix} 0 & a_{12} & \dots & a_{1N} & 1 & x_1 & y_1 \\ a_{21} & 0 & \dots & a_{2N} & 1 & x_2 & y_2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{N1} & a_{N2} & \dots & 0 & 1 & x_N & y_N \\ 1 & 1 & \dots & 1 & 0 & 0 & 0 \\ x_1 & x_2 & \dots & x_N & 0 & 0 & 0 \\ y_1 & y_2 & \dots & y_N & 0 & 0 & 0 \end{vmatrix} \begin{vmatrix} \lambda_1 \\ \lambda_2 \\ \dots \\ \lambda_N \\ v_{00} \\ v_{10} \\ v_{01} \end{vmatrix} = \begin{vmatrix} r_1 \\ r_2 \\ \dots \\ r_N \\ 0 \\ 0 \\ 0 \end{vmatrix}, \quad (2.115)$$

где $a_{ii} = G_{2,2}(0) = 0$, $a_{ij} = a_{ji} = \|P_i - P_j\|^2 \ln \|P_i - P_j\|$. Решив систему, следует восстановить сплайн-функцию по формуле

$$\sigma(x, y) = \frac{1}{2} \sum_{i=1}^N \lambda_i [(x - x_i)^2 + (y - y_i)^2] \ln [(x - x_i)^2 + (y - y_i)^2] + v_{00} + v_{10}x + v_{01}y. \quad (2.116)$$

При подсчете значения $\sigma(x, y)$ в интерполяционной точке (x_i, y_i) или очень близкой к ней нельзя пользоваться непосредственно формулой (2.116). Однако особенность в (2.116), конечно, устранима, следует просто заменить соответствующий член суммы нулем. Это относится к случаю четного n .

Отметим еще один вычислительный аспект, возникающий при решении системы. Если значения координат точек P_i заданы в реальных единицах измерения (метрах, километрах и т. д.), то величины коэффициентов a_{ij} в матрице могут весьма сильно колебаться. Это может привести к значительным ошибкам округления при решении системы методом Гаусса, так как аддитивные операции с величинами разных порядков всегда опасны для реализации на ЭВМ с ограниченной мантисой. Выход прост. Следует выполнить линейную замену переменных, переводящую область, в которой расположены интерполяционные точки, например, в единичный квадрат (или куб для $n = 3$). Точнее, подсчитаем

$$x_{\max} = \max_{i=\overline{1,N}} x_i, \quad x_{\min} = \min_{i=\overline{1,N}} x_i,$$

$$y_{\max} = \max_{i=\overline{1,N}} y_i, \quad y_{\min} = \min_{i=\overline{1,N}} y_i$$

и сделаем замену переменных

$$\begin{aligned} x' &= [x - x_{\min})/H, \quad y' = (y - y_{\min})/H, \\ H &= \max(x_{\max} - x_{\min}, y_{\max} - y_{\min}). \end{aligned} \quad (2.117)$$

После такой замены мы ни разу не наблюдали эффектов численной неустойчивости как в двумерных, так и в трехмерных задачах.

Не возникает теоретических сложностей при построении аппроксимаций не только по значениям функции в узлах хаотической сетки, но и по значениям производных (наклонов) или вторых производных (кривизн) в таких точках. Проблема в одном — необходимо решать системы линейных алгебраических уравнений с плотными матрицами, хранение которых затруднено из-за недостатка оперативной памяти ЭВМ. Один из возможных выходов — метод конечных элементов для приближения самого сплайна. Это позволяет свести систему к ленточной. Подробно эти вопросы будут рассмотрены в гл. 3. И все-таки аналитическое решение предпочтительнее, когда N невелико ($N \approx 150$). Пример расчета аналитического сплайна приведен в гл. 6, см. рис. 10, 11, 12.

После того, как коэффициенты λ_i, ν_i найдены, мы можем восполнять сплайн-функцию по формулам типа (2.116) в любой точке. Если количество точек, в которых нужно сосчитать сплайн, велико (как в задаче построения изолиний), следует воспользоваться простым приемом: пересчитать значения сплайна в узлы прямоугольной сетки, накрывающей нужную Вам зону. На прямоугольных сетках существуют простые способы интерполяции, например бикубические сплайны, расчет которых в точке сравнительно прост.

Какому же классу гладкости принадлежат интерполяционные сплайны $\sigma(P)$, построенные по формуле (2.114) на хаотической сетке в n -мерной области с использованием энергетического оператора ∇^m ? Простой анализ показывает, что $\sigma(P) \in C^k(\Omega)$, если $k < 2m - n$. Например, при $n = m = 2$ $\sigma(P) \in C^1(\Omega)$. Более тонкие рассуждения [21] показывают, что $\sigma(P)$ принадлежит пространству Соболева — Слободецкого [7] $W_2^{2m-n/2-\varepsilon}$ для любого $\varepsilon > 0$.

Отметим, что подобно одномерному случаю сплайн имеет особенности в производных только в интерполяционных точках, во всех других точках области он бесконечно дифференцируем.

Глава 3

МЕТОД КОНЕЧНЫХ ЭЛЕМЕНТОВ ПРИ ПОСТРОЕНИИ СПЛАЙН-ФУНКЦИЙ

Отметим некоторые условия, при которых в гл. 2 удалось построить точное решение вариационной задачи о поиске сплайна:

— во всех рассмотренных нами ситуациях для построения сплайнов необходимо было знать функцию Грина энергетического оператора T^*T ; это касается как одномерных задач, так и в еще большей степени многомерных;

— всюду оператор A трактовался как некоторый набор функционалов; если же A действительно оператор (например, оператор следа функции на некоторые многообразия меньшей размерности, кривые, поверхности), то приведенные в гл. 2 методы неприменимы вовсе;

— как мы видели, даже в простейших случаях (плоская область, система произвольно расположенных узлов) сплайн уже не является кусочно-полиномиальным, содержит логарифмы, дробные степени, представляется в виде суммы, число слагаемых которой растет с ростом N . Между тем преимущество сплайн-функций в конкретных вычислениях всегда состояло в простоте их локального представления;

— наконец, при расчете сплайн-функций многих переменных на хаотических сетках часто возникает необходимость решать линейные алгебраические системы с плотными матрицами, что существенно ограничивает вычислительные возможности, из-за нехватки оперативной памяти ЭВМ в особенности.

Исходя из всех этих соображений и появляется идея приближать сплайн-функцию некоторыми полиномиальными конечными элементами, искать простое приближение к сплайну, а не сам сплайн. В этом случае следует ожидать большей универсальности алгоритмов, возникновения ленточных систем алгебраических уравнений. При-

ближенная сплайн-функция всегда будет кусочно-полиномиальной, а значит, простой в употреблении. Об этом и пойдет речь в данной главе.

§ 3.1. ИНТЕРПОЛЯЦИОННЫЕ СПЛАЙНЫ НА ПОДПРОСТРАНСТВАХ

Пусть опять X , Y и Z — гильбертовы пространства, $T: X \rightarrow Y$, $A: X \rightarrow Z$ — линейные ограниченные операторы, z — фиксированный элемент пространства Z . Рассмотрим задачу интерполирования, т. е. нахождения элемента σ такого, что

$$A\sigma = z, \quad \|\sigma\|_Y^2 = \min_{x \in A^{-1}(z)} \|Tx\|_Y^2. \quad (3.1)$$

Задача (3.1) имеет единственное решение в условиях теорем существования и единственности, сформулированных в гл. 1. Среди прочих условий нам требуется, чтобы равенство $A\sigma = z$ было непротиворечиво, другими словами, полный прообраз $A^{-1}(z)$ был не пуст. Избавимся от этого требования с помощью наименьших квадратов. Заменяем задачу (3.1) следующей:

$$\|A\sigma - z\|_Z^2 = \min_{x \in X} \|Ax - z\|_Z^2, \quad (3.2)$$

и при выполнении (3.2)

$$\|\sigma\|_Y^2 = \min \|Tx\|_Y^2. \quad (3.3)$$

Очевидно, что задачу (3.2)—(3.3) можно переписать в виде

$$A^*A\sigma = A^*z, \quad \|\sigma\|_Y^2 = \min_{x \in (A^*A)^{-1}(A^*z)} \|Tx\|_Y^2, \quad (3.4)$$

где $A^*: Z \rightarrow X$ сопряжен к A . При этом полный прообраз элемента A^*z при отображении $A^*A: X \rightarrow X$ всегда не пуст. В самом деле, если предположить замкнутость образа $R(A)$ оператора A в пространстве Z , то на замкнутом линейном подпространстве $AX = R(A)$ в Z всегда реализуется минимум расстояния от точки $z \in Z$ до $R(A)$ и задача о наименьших квадратах имеет решение, единственное с точностью до элементов $N(A)$ ядра оператора

А. Далее, задача (3.4) имеет единственное решение в предположении замкнутости образов операторов T и A и условия $N(T) \cap N(A) = \{\theta_x\}$. Ядра $N(A^*A)$ и $N(A)$ совпадают. Включение $N(A^*A) \supset N(A)$ тривиально. Напротив, если $A^*Ax = \theta_x$, то, умножая это равенство скалярно на x , получим

$$(A^*Ax, x)_X = (Ax, Ax)_Z = \|Ax\|_Z^2 = 0,$$

т. е. $x \in N(A)$. Наконец, оператор A^* имеет замкнутый образ [9]. Будем в дальнейшем предполагать выполненными требования, гарантирующие существование и единственность решения задачи (3.4).

Рассмотрим теперь в X систему конечномерных подпространств $\{E_k\}_{k \geq 0}$. Будем говорить, что $\{E_k\}$ сходятся к X ($E_k \rightarrow X$), если для любого $x \in X$ найдется последовательность $x_k \in E_k$, сходящаяся к x (*).

На каждом E_k поставим задачу отыскания «интерполяционного сплайна σ_k на подпространстве» из условий

$$\|A\sigma_k - z\|_Z^2 = \min_{E_k}, \quad (3.5)$$

$$\|T\sigma_k\|_Y^2 = \min_{E_k}, \quad k = 0, 1, 2, \dots$$

Задачи (3.5) имеют единственные решения σ_k , так как пространства E_k конечномерные и требования замкнутости тривиально выполнены. Если обозначить через A_k и T_k сужения операторов T и A на подпространство E_k , то очевидно

$$N(A_k) \cap N(T_k) = N(A) \cap N(T) \cap E_k \subset N(A) \cap N(T) = \{\theta_x\}.$$

Прежде чем сформулировать теорему сходимости сплайнов на подпространстве к точному сплайну, докажем две леммы.

Лемма 1. Пусть $T: X \rightarrow Y$ и $A: X \rightarrow Z$ — линейные операторы, ядра которых удовлетворяют условию $N(T) \cap N(A) = \{\theta_x\}$. Тогда для любой последовательности $x_n \in N(T)$ из условия $\|x_n\|_X \rightarrow \infty$ следует $\|Ax_n\|_Z \rightarrow \infty$.

Доказательство. Произведем разложение элемента $x_n \in N(T)$ по ядру $N(A)$ и ортогональному дополнению $N(A)^\perp$:

$$x_n = x_{n,1} + x_{n,2}, \quad x_{n,1} \in N(A), \quad x_{n,2} \in N(A)^\perp.$$

Так как линейные пространства $N(A)$ и $N(T)$ пересекают-

*) Аналогично может быть введено понятие слабо сходящейся системы подпространств.

ся только по нулю, x_n имеет возрастающую по норме проекцию на $N(A)^\perp$, $\|x_{n,2}\|_X \rightarrow \infty$. Рассмотрим сужение $\tilde{A}$ оператора A на $N(A)^\perp$. Он осуществляет взаимно-однозначное соответствие между $N(A)^\perp$ и $R(A)$ и имеет ограниченный обратный оператор A^+ (его обычно называют псевдообратным к A). Ясно, что $x_{n,2} = A^+Ax_n$ или

$$\|x_{n,2}\|_X \leq \|A^+\| \cdot \|Ax_n\|_Z,$$

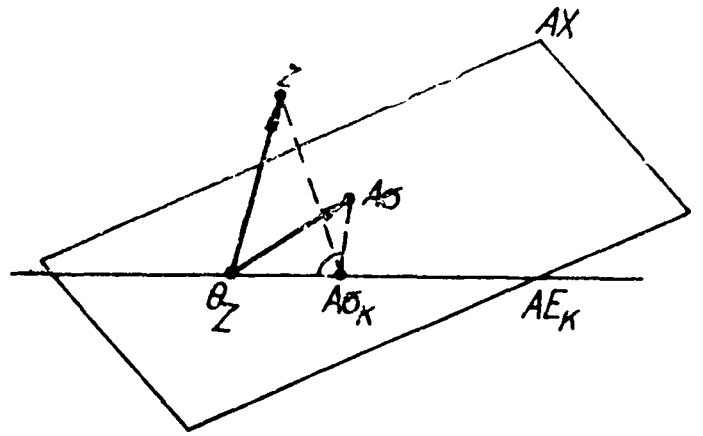


Рис. 3.

откуда и следует справедливость леммы 1.

Лемма 2. Решение σ_κ задачи (3.5) не изменяется, если вместо z поставить $A\sigma$, где σ — решение глобальной задачи (3.2)—(3.3).

Доказательство. Рассмотрим в пространстве Z замкнутое по условию подпространство $AX = R(A)$. В точке $A\sigma$ реализуется минимальное расстояние от точки $z \in Z$ до AX , т. е. $A\sigma$ — ортогональная проекция вектора z на подпространство AX . Аналогично вектор $A\sigma_\kappa$ — ортогональная проекция вектора z на AE_κ . Очевидно, что ортогональная проекция вектора $A\sigma$ на AE_κ также равна $A\sigma_\kappa$ (рис. 3). Итак, векторы z и $A\sigma$ имеют равные проекции на AE_κ . Это и означает, что множества решений задачи

$$\|Au_\kappa - z\|_Z^2 = \min_{u_\kappa \in E_\kappa} \quad (3.6)$$

и задачи

$$\|Au_\kappa - A\sigma\|_Z^2 = \min_{u_\kappa \in E_\kappa} \quad (3.7)$$

совпадают.

Теорема 3.1. Если $E_\kappa \rightarrow X$, то $\sigma_\kappa \rightarrow \sigma$ в норме пространства X .

Доказательство. Из доказательства леммы 2 очевидно, что $A\sigma_\kappa$ — проекция $A\sigma$ на AE_κ . По предположению $E_\kappa \rightarrow X$. Тогда $AE_\kappa \rightarrow AX$ в норме Z и $A\sigma_\kappa \rightarrow A\sigma$. Если обозначить

$$\begin{aligned} x_\kappa &= A^+A\sigma_\kappa, \\ x &= A^+A\sigma, \end{aligned} \quad (3.8)$$

то $x_\kappa \rightarrow x$ в норме пространства X . Далее, множество ре-

шений глобальной задачи о наименьших квадратах (3.2) может быть записано в виде $x + N(A)$, а для задачи (3.5) в виде $x_\kappa + N(A) \cap E_\kappa$. Эти множества являются замкнутыми аффинными многообразиями. Подействуем на них оператором T . Получим

$$\begin{aligned} G_\kappa &= Tx_\kappa + T(N(A) \cap E_\kappa) = y_\kappa + F_\kappa, \\ G &= Tx + TN(A) = y + F. \end{aligned} \quad (3.9)$$

Это снова замкнутые аффинные многообразия, но уже в Y . На элементах $T\sigma_\kappa$ и $T\sigma$ реализуются минимумы расстояний от нуля θ_Y пространства Y до G_κ и G соответственно. При этом

$$\begin{aligned} G_\kappa &= y_\kappa + F_\kappa, \\ G &= y + F \end{aligned} \quad (3.10)$$

и известно, что $y_\kappa \rightarrow y$, $F_\kappa \rightarrow F$. Точки минимума расстояний от θ_Y до G_κ сходятся в этом случае к точке минимума расстояния от θ_Y до G .

Итак, $T\sigma_\kappa \rightarrow T\sigma$ в норме пространства Y . Дальнейшие рассуждения стандартны. Разложим элементы σ_κ и σ следующим образом:

$$\begin{aligned} \sigma_\kappa &= \sigma_\kappa^1 + \sigma_\kappa^2, \quad \sigma_\kappa^1 \perp N(T), \quad \sigma_\kappa^2 \in N(T), \\ \sigma &= \sigma^1 + \sigma^2, \quad \sigma^1 \perp N(T), \quad \sigma^2 \in N(T). \end{aligned} \quad (3.11)$$

Легко показать, что $\sigma_\kappa^1 \rightarrow \sigma^1$. Для этого нужно рассмотреть сужение оператора T на $N(T)^\perp$; по теореме Банаха существует ограниченный обратный оператор T^+ и

$$\sigma_\kappa^1 = T^+ T \sigma_\kappa \rightarrow T^+ T \sigma = \sigma^1. \quad (3.12)$$

Последовательность σ_κ^2 ограничена. В самом деле, если последовательность $\|\sigma_\kappa^2\|_X \rightarrow \infty$, то в силу леммы 1 $\|A\sigma_\kappa^2\|_Z \rightarrow \infty$. С другой стороны $A\sigma_\kappa \rightarrow A\sigma$ и, конечно, ограничена. Кроме того, $A\sigma_\kappa^1 \rightarrow A\sigma^1$ и тоже ограничена, следовательно, должна быть ограничена и последовательность $A\sigma_\kappa^2 = A\sigma_\kappa - A\sigma_\kappa^1$, а это не так.

Из ограниченной последовательности конечномерного ядра $N(T)$ выделим сходящуюся $\sigma_\kappa^2 \rightarrow \sigma_*^2$. Ясно, что

$$\begin{aligned} T(\sigma^1 + \sigma_*^2) &= T(\sigma^1 + \sigma^2) = T\sigma, \\ A(\sigma^1 + \sigma_*^2) &= A(\sigma^1 + \sigma^2) = A\sigma \end{aligned} \quad (3.13)$$

и в силу единственности решения $\sigma^1 + \sigma_*^2 = \sigma^1 + \sigma^2 = \sigma$.

Теорема доказана.

Найдем теперь явно линейную алгебраическую систему, которую следует решить для нахождения интерполяционного сплайна σ_κ на подпространстве E_κ .

Выберем в E_κ базис из элементов $\omega_1, \omega_2, \dots, \omega_\kappa$. Решение σ_κ разложим по этому базису:

$$\sigma_\kappa = \sum_{i=1}^{\kappa} \sigma_\kappa^i \omega_i.$$

Перепишем задачу

$$\begin{aligned} A_\kappa^* A_\kappa \sigma_\kappa &= A_\kappa^* z, \\ \|T\sigma_\kappa\|_Y^2 &= \min_{E_\kappa}, \end{aligned} \tag{3.14}$$

подставив в нее разложение σ_κ по базису:

$$\begin{aligned} \sum_{i=1}^{\kappa} \sigma_\kappa^i A_\kappa^* A_\kappa \omega_i &= A_\kappa^* z, \\ \sum_{i,j=1}^{\kappa} \sigma_\kappa^i \sigma_\kappa^j (T\omega_i, T\omega_j)_Y &= \min_{E_\kappa}. \end{aligned}$$

Умножая первое равенство последовательно на $\omega_1, \omega_2, \dots, \omega_\kappa$, получим

$$\sum_{i=1}^{\kappa} \sigma_\kappa^i (A_\kappa \omega_i, A_\kappa \omega_j)_Z = (z, A_\kappa \omega_j)_Z, \quad j = 1, 2, \dots, \kappa,$$

$$\sum_{i,j=1}^{\kappa} \sigma_\kappa^i \sigma_\kappa^j (T\omega_i, T\omega_j)_Y = \min_{E_\kappa}.$$

Обозначим

$$\begin{aligned} \bar{\sigma} &= (\sigma_\kappa^1, \sigma_\kappa^2, \dots, \sigma_\kappa^\kappa)^T, \\ \bar{f} &= ((z, A_\kappa \omega_1)_Z, \dots, (z, A_\kappa \omega_\kappa)_Z)^T \end{aligned}$$

и введем матрицы $\bar{T}$ и $\bar{A}$, составленные соответственно из элементов $(T\omega_i, T\omega_j)_Y$ и $(A_\kappa \omega_i, A_\kappa \omega_j)_Z$. Тогда задача запишется в виде

$$\bar{A}\sigma = \bar{f}, \quad (\bar{T}\bar{\sigma}, \bar{\sigma}) = \min,$$

где символом $(,)$ обозначено обычное скалярное произведение векторов. Эта задача, так же как и исходная (3.14), есть задача на условный минимум квадратичного функционала $(\bar{T}\bar{\sigma}, \bar{\sigma})$ при линейных ограничениях $\bar{A}\sigma = \bar{f}$. Введем $\bar{\Lambda} = (\Lambda_1, \Lambda_2, \dots, \Lambda_\kappa)$ — вектор множителей Лаг-

ранжа. Функция Лагранжа имеет вид

$$\Phi(\bar{\sigma}, \bar{\Lambda}) = (\bar{T}\bar{\sigma}, \bar{\sigma}) + (\bar{\Lambda}, \bar{A}\bar{\sigma} - \bar{f}),$$

а условие ее минимума записывается в виде блочной системы линейных алгебраических уравнений

$$\begin{vmatrix} \bar{T} & \bar{A} \\ \bar{A} & 0 \end{vmatrix} \begin{vmatrix} \bar{\sigma} \\ \bar{\Lambda} \end{vmatrix} = \begin{vmatrix} 0 \\ \bar{f} \end{vmatrix}. \quad (3.15)$$

Система (3.15) всегда совместна, симметрична, вообще говоря, вырождена и имеет собственные числа разных знаков. Несмотря на вырожденность, у нее существует единственное решение по вектору $\bar{\sigma}$ (это следует из единственности решения исходной задачи (3.15)) и не единственное по вектору $\bar{\Lambda}$. Для отыскания $\bar{\sigma}$ достаточно найти любое решение системы (3.15). Это можно сделать с помощью широкого класса итерационных методов [10]. Весьма удобным является использование метода минимальных итераций Ланцоша [11].

§ 3.2. СГЛАЖИВАЮЩИЕ СПЛАЙНЫ НА ПОДПРОСТРАНСТВАХ

Займемся теперь приближенным решением задачи сглаживания

$$\Phi_{\alpha}(\sigma_{\alpha}) = \alpha \|T\sigma_{\alpha}\|_Y^2 + \|A\sigma_{\alpha} - z\|_Z^2 = \min_X. \quad (3.16)$$

Будем опять предполагать условия теорем существования и единственности выполненными, ядро $N(T)$ конечномерным. Как и в § 3.1, $\{E_k\}_{k \geq 0}$ — система конечномерных подпространств в X .

Сглаживающим сплайном σ_{α}^k на подпространстве E_k назовем решение задачи

$$\alpha \|T\sigma_{\alpha}^k\|_Y^2 + \|A\sigma_{\alpha}^k - z\|_Z^2 = \min_{E_k}. \quad (3.17)$$

В указанных предположениях решение задачи (3.17) существует и единственно.

Теорема 3.2. Если последовательность подпространств $E_k \subset E_{k+1}$ слабо сходится к X , то последовательность сглаживающих сплайнов на подпространствах σ_{α}^k слабо сходится в X к сглаживающему сплайну σ_{α} при $k \rightarrow \infty$.

Доказательство. Докажем сначала, что последовательность $\{\sigma_\alpha^\kappa\}$ ограничена равномерно по индексу κ . Для этого представим, как обычно, элемент σ_α^κ в виде

$$\sigma_\alpha^\kappa = \sigma_\alpha^{\kappa,1} + \sigma_\alpha^{\kappa,2}, \quad (3.18)$$

где $\sigma_\alpha^{\kappa,1} \perp N(T)$, $\sigma_\alpha^{\kappa,2} \in N(T)$. Так как все пространства E_κ содержат нулевую точку θ_x , мы можем записать неравенство

$$\frac{1}{\alpha} \Phi_\alpha(\sigma_\alpha^\kappa) = \|T\sigma_\alpha^\kappa\|_Y^2 + \frac{1}{\alpha} \|A\sigma_\alpha^\kappa - z\|_Z^2 \leq \frac{1}{\alpha} \Phi_\alpha(\theta_x) = \frac{1}{\alpha} \|z\|_Z^2. \quad (3.19)$$

Аналогично доказательству теоремы 3.1 легко получаем, что последовательность $\{\sigma_\alpha^{\kappa,1}\}$ ограничена по κ , так как $\|T\sigma_\alpha^{\kappa,1}\|_Y$ ограничена константой, не зависящей от κ . Далее, используя лемму 2, немедленно получим ограниченность $\{\sigma_\alpha^{\kappa,2}\}$.

Выделим из ограниченной последовательности $\{\sigma_\alpha^\kappa\}$ подпоследовательность $\sigma_\alpha^{\kappa'}$, слабо сходящуюся в X к σ_* . Употребляя условие ортогональности для сглаживающих сплайнов $\sigma_\alpha^{\kappa'}$, мы можем записать

$$\alpha(T\sigma_\alpha^{\kappa'}, T\omega_l)_Y + (A\sigma_\alpha^{\kappa'} - z, A\omega_l - z)_Z = -(A\sigma_\alpha^{\kappa'} - z, z)_Z. \quad (3.20)$$

Здесь ω_l — любая последовательность элементов пространства X такая, что $\omega_l \in E_l$ и $\kappa' \geq l$. Переходя в (3.20) к пределу при $\kappa' \rightarrow \infty$, получим

$$\alpha(T\sigma_*, T\omega_l)_Y + (A\sigma_* - z, A\omega_l - z)_Z = -(A\sigma_* - z, z)_Z. \quad (3.21)$$

Если в качестве ω_l взять последовательность, слабо сходящуюся к σ_α (это можно сделать, так как $E_\kappa \xrightarrow{c} X$), то при $l \rightarrow \infty$ из (3.21) следует

$$\alpha(T\sigma_*, T\sigma_\alpha)_Y + (A\sigma_* - z, A\sigma_\alpha - z)_Z = -(A\sigma_* - z, z)_Z. \quad (3.22)$$

С другой стороны, левая часть равенства (3.22) в силу условия ортогональности (1.24) равна $-(A\sigma_\alpha - z, z)_Z$. Используя теперь в качестве ω_l последовательность, слабо сходящуюся к σ_* , из (3.21) заключаем, что

$$\alpha\|T\sigma_*\|_Y^2 + \|A\sigma_* - z\|_Z^2 = -(A\sigma_* - z, z)_Z. \quad (3.23)$$

Отсюда, как легко заметить, следует, что

$$\Phi_{\alpha}(\sigma_{*}) = \Phi_{\alpha}(\sigma_{\alpha}) \quad (3.24)$$

и в силу единственности решения задачи (3.16) $\sigma_{*} = \sigma_{\alpha}$. Итак, любая слабая предельная точка последовательности $\{\sigma_{\alpha}^{\kappa}\}$ совпадает с σ_{α} . Теорема доказана.

Следствие. Пусть теперь оператор A обладает конечномерным образом $R(A)$. Докажем, что в этом случае σ_{α}^{κ} сходится к σ_{α} по норме X .

Действительно,

$$\Phi_{\alpha}(\sigma_{\alpha}^{\kappa}) = \alpha \|T\sigma_{\alpha}^{\kappa}\|_Y^2 + \|A\sigma_{\alpha}^{\kappa} - z\|_Z^2 = - (A\sigma_{\alpha}^{\kappa} - z, z)_Z$$

в силу теоремы 3.2 стремится к

$$- (A\sigma_{\alpha} - z, z)_Z = \alpha \|T\sigma_{\alpha}\|_Y^2 + \|A\sigma_{\alpha} - z\|_Z^2 = \Phi_{\alpha}(\sigma_{\alpha})$$

при $\kappa \rightarrow \infty$. Так как $R(A)$ конечномерно, $A\sigma_{\alpha}^{\kappa} \rightarrow A\sigma_{\alpha}$ сильно и $\|A\sigma_{\alpha}^{\kappa} - z\|_Z \rightarrow \|A\sigma_{\alpha} - z\|_Z$. Тогда, очевидно, $\|T\sigma_{\alpha}^{\kappa}\|_Y \rightarrow \|T\sigma_{\alpha}\|_Y$, а это, учитывая слабую сходимость $T\sigma_{\alpha}^{\kappa}$ к $T\sigma_{\alpha}$, влечет сильную сходимость $T\sigma_{\alpha}^{\kappa}$ к $T\sigma_{\alpha}$. Повторяя стандартные рассуждения, приходим к

$$\|\sigma_{\alpha}^{\kappa} - \sigma_{\alpha}\|_X \rightarrow 0, \quad \kappa \rightarrow \infty. \quad (3.25)$$

Получим теперь явно системы линейных алгебраических уравнений, которые необходимо решить для нахождения сглаживающего сплайна σ_{α}^{κ} на подпространстве E_{κ} .

Пусть $\omega_1, \omega_2, \dots, \omega_{\kappa}$ — базис E_{κ} . Разложим по базису искомый элемент σ_{α}^{κ} :

$$\sigma_{\alpha}^{\kappa} = \sum_{i=1}^{\kappa} \sigma^{(i)} \omega_i. \quad (3.26)$$

Подставим выражение (3.26) в функционал, который нужно минимизировать,

$$\begin{aligned} \Phi_{\alpha}(\sigma_{\alpha}^{\kappa}) &= \alpha \sum_{i,j=1}^{\kappa} \sigma^{(i)} \sigma^{(j)} (T\omega_i, T\omega_j)_Y + \sum_{i,j=1}^{\kappa} \sigma^{(i)} \sigma^{(j)} (A\omega_i, A\omega_j)_Z - \\ &\quad - 2 \sum_{i=1}^{\kappa} \sigma^{(i)} (A\omega_i, z)_Z + \|z\|_Z^2. \end{aligned} \quad (3.27)$$

Обозначим $\bar{\sigma} = (\sigma^{(1)}, \sigma^{(2)}, \dots, \sigma^{(\kappa)})^T$, $\bar{f} = ((A\omega_1, z)_Z, \dots, (A\omega_{\kappa}, z)_Z)^T$. Пусть матрицы $\bar{T}$ и $\bar{A}$ составлены соответственно из элементов $(T\omega_i, T\omega_j)_Y$ и $(A\omega_i, A\omega_j)_Z$. Тогда функционал (3.27) можно переписать как функцию

вектора $\bar{\sigma}$:

$$\Phi_{\alpha}(\sigma_{\alpha}^{\kappa}) = \Phi_{\alpha}(\bar{\sigma}) = \alpha(\bar{T}\bar{\sigma}, \bar{\sigma}) + (\bar{A}\bar{\sigma}, \bar{\sigma}) - 2(\bar{f}, \bar{\sigma}) + \|z\|_Z^2. \quad (3.28)$$

Здесь символом $(,)$ обозначено обычное скалярное произведение векторов. Дифференцируя $\Phi_{\alpha}(\bar{\sigma})$ по переменным $\sigma^{(i)}$ и приравнявая производные 0, получим условие минимума квадратичного функционала $\Phi_{\alpha}(\bar{\sigma})$:

$$(\alpha\bar{T} + \bar{A})\bar{\sigma} = \bar{f}. \quad (3.29)$$

Это невырожденная линейная алгебраическая система с симметричной положительно определенной матрицей $\alpha\bar{T} + \bar{A}$, которая может быть решена с помощью широкого класса итерационных и прямых методов. Матрицы $\bar{T}$ и $\bar{A}$ совпадают с матрицами, возникающими при решении задачи интерполяции без сглаживания в § 1 этой главы.

§ 3.3. МЕТОД КОНЕЧНЫХ ЭЛЕМЕНТОВ НА ПРИМЕРЕ ДВУМЕРНОЙ ЗАДАЧИ СПЛАЙН-АППРОКСИМАЦИИ

В этом параграфе мы отвлечемся от общих функциональных постановок и сосредоточим внимание на совершенно конкретной задаче. Пусть $P_1, P_2, \dots, P_N$ — некоторая система точек на плоскости. Рассмотрим прямоугольник Ω такой, чтобы эти точки им покрывались, $P_i \in \bar{\Omega}$, $i = 1, N$. Нанесем на прямоугольник Ω некоторую квадратную сетку Ω_h , $h_x = h_y = h$. Разумеется, длины сторон прямоугольника предполагаются кратными шагу h . Каждому узлу сетки сопоставим билинейную базисную функцию $\omega_{ij}(x, y) = \omega_i(x)\omega_j(y)$, равную 1 в узле (i, j) и 0 во всех остальных. Всевозможные линейные комбинации таких функций образуют конечномерное пространство. Необходимо построить функцию $\sigma_h(x, y)$ из этого пространства, которая удовлетворяет, если это вообще возможно, интерполяционному условию $\sigma_h(P_i) = r_i$, $i = 1, N$, в противном случае доставляет минимум функционалу наименьших квадратов

$$\|A\sigma_h - r\|^2 = \sum_{i=1}^N [\sigma_h(P_i) - r_i]^2 = \min \quad (3.30)$$

и наряду с этим минимизирует функционал типа энер-

гии вида

$$\int_{\Omega} \left[\left(\frac{\partial \sigma_h}{\partial x} \right)^2 + \left(\frac{\partial \sigma_h}{\partial y} \right)^2 \right] d\Omega. \quad (3.31)$$

(3.30), (3.31) — типично сплайновая задача, только на конечномерном пространстве X , которая в соответствии с общей теорией имеет единственное решение, если множество точек P_i не пусто.

В соответствии с теорией, развитой в § 3.1, 3.2, необходимо сформировать две матрицы $\bar{T}$ и $\bar{A}$, а затем решить систему (3.15), если решается интерполяционная задача, или (3.29), если решается задача сглаживания.

Необходимо отыскать решение задачи σ_h в виде

$$\sigma_h(x, y) = \sum_{i,j} \sigma_{ij} \omega_{ij}(x, y);$$

не будем производить перенумерацию узлов сетки, чтобы сделать вектор $\bar{\sigma} = \{\sigma_{ij}\}$ одномерным. Тогда отвечающие ему матрицы $\bar{T}$, $\bar{A}$ будут четырехиндексными,

$$t_{ijkl} = (T \omega_{ij}, T \omega_{kl})_{L_2 \times L_2},$$

$$a_{ijkl} = (A \omega_{ij}, A \omega_{kl})$$

или, подробнее, с учетом $\omega_{ij}(x, y) = \omega_i(x) \omega_j(y)$,

$$\begin{aligned} t_{ijkl} = \int_{\Omega} \left[\frac{\partial \omega_i(x)}{\partial x} \omega_j(y) \frac{\partial \omega_k(x)}{\partial x} \omega_l(y) + \omega_i(x) \frac{\partial \omega_j(y)}{\partial y} \times \right. \\ \left. \times \omega_k(x) \frac{\partial \omega_l(y)}{\partial y} \right] dx dy, \\ a_{ijkl} = \sum_{m=1}^N \omega_i(x_m) \omega_j(y_m) \omega_k(x_m) \omega_l(y_m), \end{aligned} \quad (3.32)$$

где P_m имеет координаты (x_m, y_m) , $m = \overline{1, N}$. Очевидно, что

$$t_{ijkl} = d_{ik} u_{jl} + u_{ik} d_{jl}, \quad (3.33)$$

где

$$u_{ij} = \int \omega_i(x) \omega_j(x) dx, \quad d_{ij} = \int \omega'_i(x) \omega'_j(x) dx. \quad (3.34)$$

Учитывая явный вид функций $\omega_i(x)$ (рис. 4) и производя интегрирование, получим

$$\begin{aligned} u_{11} = u_{nn} = h/3, \quad d_{11} = d_{nn} = 1/h, \\ u_{ii} = 2h/3, \quad i \neq 1, \quad i \neq n, \quad d_{ii} = 2/h, \quad i \neq 1, \quad i \neq n, \\ u_{i, i \pm 1} = h/6, \quad d_{i, i \pm 1} = -1/h. \end{aligned}$$

Изобразим на рис. 5 три возможные ситуации: угловая точка (а), граничная, но не угловая точка (б) и внутренняя точка (в). Таковы возникающие коэффициенты t_{ijkl} с точностью до множителя

$1/3$, который совершенно не важен, так как исходный вариационный функционал (3.31) можно было умножить на 3, отчего решение задачи о его минимуме не изменилось бы. Все остальные коэффициенты при $|i - \kappa| > 1$ или при $|j - l| > 1$ равны 0.

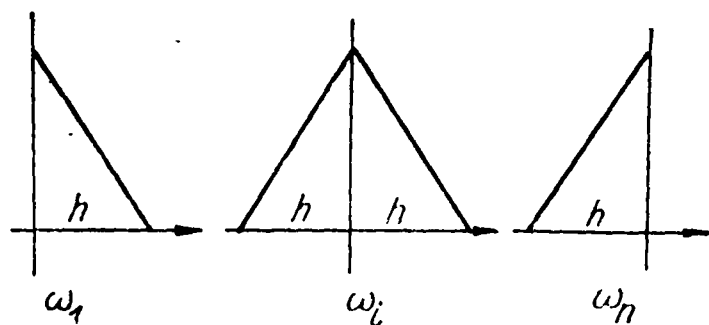


Рис. 4.

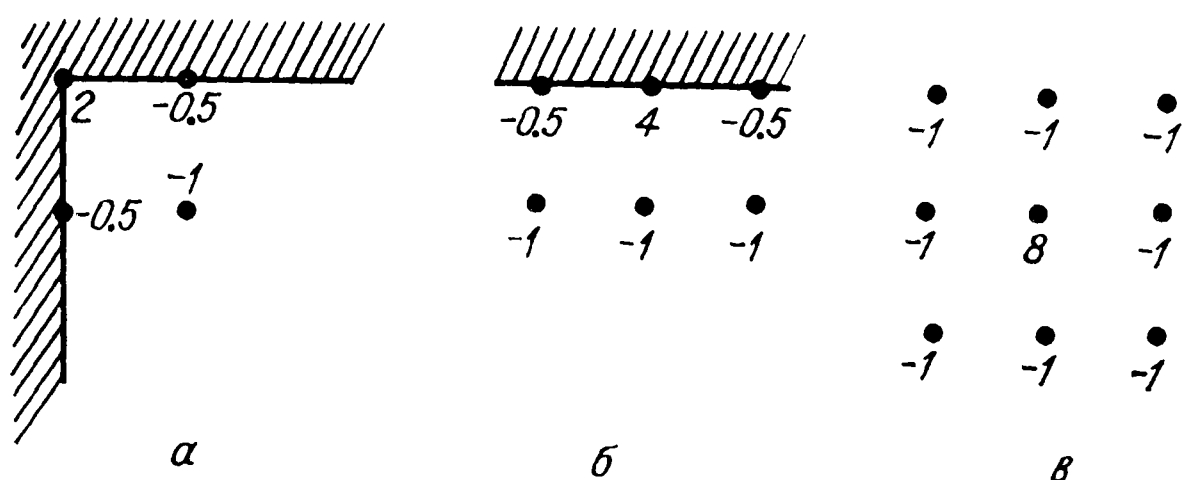


Рис. 5.

Займемся теперь коэффициентом a_{ijkl} . Ясно, что при $|i - \kappa| > 1$ или при $|j - l| > 1$ $a_{ijkl} = 0$, так как в этом случае $\omega_i(x)\omega_\kappa(x) \equiv 0$ или $\omega_j(y)\omega_l(y) \equiv 0$, поскольку носители функций не пересекаются. При суммировании в формуле (3.32) следует отбросить все точки (x_m, y_m) , не входящие в пересечение носителей функций $\omega_i(x)\omega_j(y)$ и $\omega_\kappa(x)\omega_l(y)$. Для того, чтобы не проверять каждый раз принадлежность точки P_m этому множеству, следует один раз выполнить сортировку исходной совокупности точек, т. е. расположить точки P_m так, чтобы сначала шли все, лежащие в первой ячейке сетки, а затем — во второй и т. д. Одновременно необходимо запоминать номер последней точки, лежащей в ячейке (i, j) в двумерном целочисленном массиве p_{ij} . После этого можно оперативно вести суммирование по ячейке (i, j) от p_- до p_+ , где $p_+ = p_{ij}$, p_- — номер последней точки из предыдущей ячейки плюс 1.

Произведенная сортировка позволит эффективно вычислить компоненты правой части системы

$$f_{ij} = (\bar{r}, A\omega_{ij}) = \sum_{m=1}^N r_m \omega_i(x_m) \omega_j(y_m). \quad (3.35)$$

Суммирование в (3.35) нужно вести только по точкам (x_m, y_m) , входящим в носитель функции $\omega_{ij}(x, y)$.

Написание программы расчета приближенного сплайна состоит из следующих этапов:

- написать подпрограмму линейной интерполяции в двумерном случае по значениям в узлах прямоугольной сетки, что обеспечит вычисление $\omega_{ij}(P_m)$;

- написать подпрограмму сортировки исходной совокупности точек по ячейкам сетки;

- написать процедуру умножения вектора σ_{ij} на матрицу $\bar{T}$, исходя из известного сеточного шаблона (см. рис. 5);

- написать процедуру умножения вектора σ_{ij} на матрицу $\bar{A}$ с использованием линейной интерполяции и результатов сортировки;

- имея теперь быстрые алгоритмы умножения на матрицы $\bar{T}$ и $\bar{A}$, каждая строка которых содержит не более 9 ненулевых элементов, легко написать любой итерационный процесс для решения систем (3.15) или (3.29), в котором требуется только уметь умножать на матрицу системы и не требуется ее хранить.

Описанная организация алгоритма позволяет решать системы весьма высоких порядков, сохраняя разумные требования к объему оперативной памяти. Результатом решения является массив элементов σ_{ij} , которые равны значениям искомой функции в узлах квадратной сетки. Такой пересчет позволяет в дальнейшем легко производить любые операции: интегрировать, дифференцировать, интерполировать функцию, вычерчивать поля изолиний и многое другое.

В этом параграфе мы сознательно до предела упростили ситуацию: квадратная сетка, билинейные конечные элементы, простейший энергетический функционал. Сделано это с целью объяснения необходимых этапов в программировании. Ничто не мешает использовать более гладкие конечные элементы и более сложные функционалы энергии. Это может привести к другим сеточным шаблонам для матриц $\bar{T}$ и $\bar{A}$, алгоритмы умножения на $\bar{T}$ и $\bar{A}$ станут более громоздкими, но общая структура программы не изменится.

Наконец, на практике весьма полезно считать шаги сетки по x и по y равными 1. Этого легко добиться линейной заменой переменных. Если измерять координаты точек P_i в шагах по x и по y , не понадобится произво-

дить многочисленные деления на шаг сетки при линейной интерполяции — вычислением целой части координат x_i и y_i немедленно определяется, в какой ячейке лежит точка. Такая замена ускоряет работу программ в 2—3 раза.

Глава 4

СХОДИМОСТЬ СПЛАЙН-ФУНКЦИЙ

В пределах этой главы мы докажем общую теорему сходимости сплайнов и с ее помощью получим оценки сходимости для некоторых конкретных сплайнов. Далее будет доказана теорема о сходимости сглаживающих сплайнов к интерполяционным и рассмотрена проблема выбора параметра сглаживания.

§ 4.1. ОСНОВНАЯ ТЕОРЕМА СХОДИМОСТИ

Рассмотрим некоторую, вообще говоря, бесконечную систему $A = \{A_i : X \rightarrow Z_i, i = 1, 2, \dots\}$ линейных ограниченных операторов A_i , действующих из гильбертова пространства X в гильбертово пространство $Z_i, i = 1, 2, \dots$.

Определение 1. Будем говорить, что последовательность x_n сходится к x в смысле системы операторов A ($x_n \xrightarrow{A} x$), если

$$A_i x_n \rightarrow A_i x, \quad i = 1, 2, \dots, n \rightarrow \infty. \quad (4.1)$$

Определение 2. Систему операторов A назовем правильной, если сходимость любой последовательности $x_n \xrightarrow{A} x$ в смысле системы операторов A влечет слабую сходимость x_n к x на множестве K , всюду плотном в X ,

$$[x_n \xrightarrow{A} x] \Rightarrow [\exists K, \bar{K} = X, (x_n, k)_X \rightarrow (x, k)_X \forall k \in K]. \quad (4.2)$$

Рассмотрим элемент $\varphi_* \in X$. Аппроксимируем его интерполяционными сплайнами σ_N , являющимися решениями последовательности задач

$$\begin{aligned} A_i \sigma_N &= A_i \varphi_*, \quad i = 1, 2, \dots, N, \\ \|T \sigma_N\|_Y^2 &= \min. \end{aligned} \quad (4.3)$$

Будем в дальнейшем предполагать, что условия существования и единственности сплайна σ_N выполнены (см. гл. 1).

Теорема 4.1. Если ядро оператора T имеет конечную размерность и система операторов $A = \{A_i, i = 1, 2, \dots\}$, по которой строятся сплайны σ_N , правильная, последовательность σ_N сходится по норме пространства X к элементу φ_* .

Доказательство. Очевидно, что последовательность σ_N сходится к φ_* в смысле системы операторов $A, \sigma_N \xrightarrow{A} \varphi_*$.

Действительно, какой бы мы не взяли оператор A_i при $N \geq i$ $A_i \sigma_N = A_i \varphi_*$. Так как система A правильная, σ_N слабо сходится к φ_* на всюду плотном в X множестве K . Нам потребуется

Лемма. Последовательность сплайнов σ_N ограничена по норме пространства X константой, не зависящей от N .

Доказательство леммы. Разложим пространство X в ортогональную сумму ядра T и его ортогонального дополнения

$$X = N(T)^\perp \oplus N(T). \quad (4.4)$$

Ясно, что $N(T)^\perp$ — снова гильбертово пространство со скалярным произведением пространства X . В соответствии с разложением (4.4)

$$\sigma_N = \sigma_N^1 + \sigma_N^2, \quad \sigma_N^1 \in N(T)^\perp, \quad \sigma_N^2 \in N(T).$$

Рассмотрим теперь сужение $\tilde{T}$ оператора T на $N(T)^\perp$. Оператор $\tilde{T}$ осуществляет взаимно-однозначное непрерывное отображение гильбертова пространства $N(T)^\perp$ в гильбертово пространство $TX = R(T)$ (напомним, что в соответствии с требованиями теорем существования и единственности гл. 1 образ $R(T)$ оператора T замкнут). Тогда по теореме Банаха об обратном операторе [9] оператор $\tilde{T}^{-1}$ непрерывен на $R(T)$. Отсюда, используя очевидное неравенство $\|T\sigma_N\|_Y \leq \|T\varphi_*\|_Y$, имеем

$$\|\sigma_N^1\|_X \leq \|\tilde{T}^{-1}\| \cdot \|T\sigma_N^1\|_Y = \|\tilde{T}^{-1}\| \cdot \|T\sigma_N\|_Y \leq \|\tilde{T}^{-1}\| \cdot \|T\varphi_*\|_Y. \quad (4.5)$$

Докажем теперь ограниченность последовательности σ_N^2 конечномерного ядра $N(T)$. Каждый элемент k , принадлежащий всюду плотному в X множеству K , тоже

можно разложить в соответствии с (4.4):

$$k = k^1 + k^2. \quad (4.6)$$

Из элементов k^2 такого вида можно выделить нормированный конечный базис ядра $N(T)$ $k_1^2, k_2^2, \dots, k_s^2$, $s = \dim N(T)$. Тогда

$$(\sigma_{N_2}^2 k_i^2)_X = (\sigma_{N_2}^2 k_i)_X = (\sigma_N - \sigma_{N_2}^1 k_i)_{X_2} \quad i = 1, 2, \dots, s. \quad (4.7)$$

Ясно, что $(\sigma_{N_2} k_i)_X \rightarrow (\varphi_*, k_i)$, поэтому сходящаяся последовательность (σ_N, k_i) ограничена константой, не зависящей от N , а последовательность $(\sigma_{N_2}^1 k_i)_X$ ограничена константой в силу (4.5). Окончательно

$$\|\sigma_N\|_X \leq \text{const}, \quad (4.8)$$

что и доказывает лемму.

Продолжим доказательство теоремы. Так как последовательность σ_N слабо сходится к φ_* на всюду плотном в X множестве и ограничена, она сходится слабо на всем X . [9].

Рассмотрим теперь числовую последовательность $\{\|T\sigma_N\|_Y\}$. Эта последовательность не убывает, так как $\|T\sigma_N\|_Y = \min$ при условии $A_i \sigma_N = A_i \varphi_*$ $i = 1, 2, \dots, N$, а $\|T\sigma_{N+1}\|_Y = \min$ при условии $A_i \sigma_{N+1} = A_i \varphi_*$, $i = 1, 2, \dots, N, N+1$, т.е. множество, на котором достигается минимум, во втором случае уже. Кроме того, последовательность $\|T\sigma_N\|_Y$ ограничена, следовательно, имеет конечный предел $R < \infty$, $\|T\sigma_N\|_Y \rightarrow R$ при $N \rightarrow \infty$. С другой стороны, согласно условию ортогональности интерполяционных сплайнов

$$\|T\sigma_N\|_Y^2 = (T\varphi_*, T\sigma_N)_Y. \quad (4.9)$$

Так как σ_N сходится слабо к φ_* в X , $T\sigma_N$ сходится слабо к $T\varphi_*$ в Y . Переходя в (4.9) к пределу при $N \rightarrow \infty$, получаем

$$R^2 = \|T\varphi_*\|_Y^2.$$

Итак, $T\sigma_N \xrightarrow{c} T\varphi_*$ и $\|T\sigma_N\|_Y \rightarrow \|T\varphi_*\|_Y$, а значит [9], сходится к $T\varphi_*$ сильно в Y .

Если теперь в соответствии с разложением (4.5) записать

$$\begin{aligned} \sigma_N &= \sigma_N^1 + \sigma_{N_2}^2 \\ \varphi_* &= \varphi_*^1 + \varphi_*^2, \end{aligned} \quad (4.10)$$

то очевидно, что $\sigma_N^1 = \tilde{T}^{-1} T \sigma_N$ сходится сильно в X к $\varphi_*^1 = \tilde{T}^{-1} T \varphi_*$.

Из слабой сходимости σ_N к φ_* очевидно, что $\sigma_N^2 \xrightarrow{c} \varphi_*^2$, но это — сходимость элементов конечномерного пространства — ядра $N(T)$, и она, конечно, является сильной. Окончательно, σ_N сходится к φ_* в норме пространства X . Теорема доказана.

З а м е ч а н и е. В работах А. Имамова [37, 39] показано, что требование правильности системы операторов, по которым строятся интерполяционные сплайны, является не только достаточным, но и необходимым условием сходимости интерполяционного процесса.

§ 4.2. СХОДИМОСТЬ СПЛАЙНОВ НЕЧЕТНОЙ СТЕПЕНИ НА ОТРЕЗКЕ

Исследуем сходимость кусочно-полиномиальных сплайн-интерполяций на конечном отрезке $[a, b]$. Для этого рассмотрим (см. гл. 1 и 2) гильбертовы пространства $X = W_2^n[a, b]$ и $Y = L_2[a, b]$ и линейный ограниченный оператор $T = \frac{d^n}{dx^n}: W_2^n \rightarrow L_2, n \geq 1$. Пусть S_N — последовательность сеток, причем $S_N \subset S_{N+1}$ и $\bigcup_{i=1}^{\infty} S_i$ есть всюду

плотное множество точек интервала $[a, b]$. Каждой сетке $S_N = \{a = x_1^N < x_2^N < \dots < x_{q_N}^N = b\}$ поставим в соответствие функционалы $k_{x_1^N}, k_{x_2^N}, \dots, k_{x_{q_N}^N}$, удовлетворяющие для любой функции $f \in W_2^n[a, b]$

$$(f, k_{x_i^N})_{W_2^n} = f(x_i^N). \quad (4.11)$$

Легко видеть, что эта система функционалов, соответствующая всем сеткам, образует правильную систему операторов, если рассматривать функционал как оператор из W_2^n в пространство вещественных чисел R^1 . Рассмотрим теперь последовательность задач построения интерполяционных сплайнов из условий

$$\begin{aligned} \sigma_N(x_i^N) &= \varphi_*(x_i^N), \quad i = 1, 2, \dots, q_N, \\ \int_a^b \sigma_N^{(n)^2}(x) dx &= \min. \end{aligned} \quad (4.12)$$

На основании теоремы 1.5 гл. 1, решение таких задач существует и единственно при $q_N \geq n$, а из гл. 2 известно, что σ_N — кусочно-полиномиальная функция степени $2n - 1$. Получим оценки сходимости σ_N к точной функции $\varphi_* \in W_2^n[a, b]$.

Пусть $\psi_N = \varphi_* - \sigma_N$ и N — такое число, что

$$\|S_N\| = \max_i |x_{i+1}^N - x_i^N| \leq h. \quad (4.13)$$

Так как в интервале $A = [x_i^N, x_{i+n}^N]$ функция $\psi_N(x)$ имеет, по крайней мере, $n + 1$ корень (в интерполяционных узлах), то по теореме Ролля $\psi_N'(x)$ имеет n корней, $\psi_N''(x)$ — $(n - 1)$ корней и т. д., $\psi_N^{(n-1)}(x)$ — по крайней мере, 1 корень. Обозначим через ζ_0 некоторый корень ψ_N в A , через ζ_1 — корень ψ_N' в A , через ζ_{n-1} — корень $\psi_N^{(n-1)}$. Тогда можно записать разложения Тейлора с остаточным членом в интегральной форме:

$$\begin{aligned} \psi_N(x) &= \psi_N'(\zeta_0)(x - \zeta_0) + \dots + \psi_N^{(n-1)}(\zeta_0) \frac{(x - \zeta_0)^{n-1}}{(n-1)!} + \\ &\quad + \int_{\zeta_0}^x \frac{(x-t)^{n-1}}{(n-1)!} \psi_N^{(n)}(t) dt, \\ \psi_N'(x) &= \psi_N''(\zeta_1)(x - \zeta_1) + \dots + \psi_N^{(n-1)}(\zeta_1) \frac{(x - \zeta_1)^{n-2}}{(n-2)!} + \\ &\quad + \int_{\zeta_1}^x \frac{(x-t)^{n-2}}{(n-2)!} \psi_N^{(n)}(t) dt, \\ &\dots \end{aligned} \quad (4.14)$$

$$\psi_N^{(n-1)}(x) = \int_{\zeta_{n-1}}^x \psi_N^{(n)}(t) dt.$$

Согласно теореме сходимости 4.1

$$\int_a^b \psi_N^{(n)^2}(t) dt = o(1). \quad (4.15)$$

Тогда используя последнее из равенств (4.14) и неравенство Коши, получим

$$\begin{aligned} |\psi_N^{(n-1)}(x)| &= \left| \int_{\xi_{n-1}}^x \psi_N^{(n)}(t) dt \right| \leq \\ &\leq \left| \left(\int_{\xi_{n-1}}^x dt \right) \right|^{1/2} \left(\int_a^b \psi_N^{(n)2}(t) dt \right)^{1/2} = o(h^{1/2}). \end{aligned} \quad (4.16)$$

Используя неравенство (4.16) в предпоследнем из разложений (4.14), аналогично получим

$$|\psi_N^{(n-2)}(x)| = o(h^{3/2}). \quad (4.17)$$

Последовательно применив все равенства (4.14), приходим к следующим оценкам сходимости самих сплайн-функций и их производных:

$$\|\sigma_N^{(i)} - \varphi_*^{(i)}\|_{C[a,b]} = o(h^{n-i-1/2}) \quad (4.18)$$

для $i = 0, 1, \dots, n-1$. В частности, при $n=2$ кубическая сплайн-интерполяция сходится к $\varphi_* \in W_2^2[a, b]$ с порядком $o(h^{3/2})$, а ее первая производная σ_N' — с порядком $o(h^{1/2})$ в норме $C[a, b]$, вторая производная σ_N'' — с порядком $o(1)$ в норме $L_2[a, b]$.

§ 4.3. СХОДИМОСТЬ СПЛАЙНОВ В ДВУМЕРНОЙ ОБЛАСТИ НА СГУЩАЮЩЕЙСЯ ХАОТИЧЕСКОЙ СЕТКЕ

В § 2.7 гл. 2 мы построили аналитическое решение $\sigma(P)$ задачи интерполяции в n -мерной области Ω по системе узлов $P_1, P_2, \dots, P_N$ из условий

$$\sigma(P_i) = r_i, \quad i = \overline{1, N},$$

$$\int_{\Omega} \sum_{|\alpha|=m} \frac{m!}{\alpha!} (D^\alpha \sigma)^2 d\Omega = \min.$$

Положим $n = m = 2$ и рассмотрим последовательность хаотических сеток

$$S_i = \{P_1^{(i)}, P_2^{(i)}, \dots, P_{N_i}^{(i)}\}, \quad i = 1, 2, \dots \quad (4.19)$$

Каждой сетке сопоставим ее характерный параметр

$$\varepsilon_i = \sup_{P \in \Omega} \min_{\kappa} \|P - P_\kappa^{(i)}\|_{2_i} \quad (4.20)$$

где $\|\cdot\|_2$ — евклидово расстояние в R^n . Ясно, что сетка S_i

образует ε -сеть в области Ω при $\varepsilon \geq \varepsilon_i$ и не образует ε -сети при $\varepsilon < \varepsilon_i$. Такое критическое значение параметра ε -сети является аналогом шага сетки в одномерном случае и может служить мерой «густоты» хаотической сетки.

Предположим теперь, что $\varepsilon_i \rightarrow 0$ при $i \rightarrow \infty$. Это означает, что множество

$$S = \bigcup_{i=1}^{\infty} S_i \quad (4.21)$$

является всюду плотным в Ω . Тогда, очевидно, элементы $\{k_a\}$, $a \in S$, такие, что $(k_a, \varphi)_{W_2^2} = \varphi(a)$ для любой $\varphi \in W_2^2$, $a \in S$, как функционалы, представляют собой правильную систему (см. § 4.1). Следовательно, мы можем воспользоваться общей теоремой сходимости.

Пусть $\varphi_* \in W_2^2(\Omega)$ — некоторая фиксированная функция. Будем интерполировать ее последовательно на сетках $S_1, S_2, \dots, S_N, \dots$ решениями $\sigma_N(P)$ задач

$$\sigma_N|_{S_N} = \varphi_*|_{S_N},$$

$$\int_{\Omega} \sum_{|\alpha|=2} \frac{2!}{\alpha!} (D^{\alpha} \sigma_N)^2 d\Omega = \min.$$

На основании общей теоремы сходимости (§ 4.1) при $|\alpha| = 2$ имеем

$$\|D^{\alpha} \sigma_N - D^{\alpha} \varphi_*\|_{L_2(\Omega)} \rightarrow 0, \quad N \rightarrow \infty. \quad (4.22)$$

Используем этот результат для получения оценки сходимости сплайнов к точной функции. Обозначим через Ω_{δ} совокупность точек области Ω , удаленных от границы Γ области Ω на расстояние, превышающее $\delta > 0$,

$$\Omega_{\delta} = \{P \in \Omega : \rho(P, \Gamma) > \delta\}, \quad (4.23)$$

$$\rho(P, \Gamma) = \inf_{Q \in \Gamma} \|P - Q\|_2. \quad (4.24)$$

Теорема 4.2. В любой δ -внутренности Ω_{δ} области Ω справедлива оценка

$$\|\sigma_N - \varphi_*\|_{C(\Omega_{\delta})} = o(\varepsilon_N^{1-\alpha}), \quad (4.25)$$

где $\alpha > 0$ — сколь угодно малое фиксированное число.

Доказательство. Употребим известную [9] интегральную формулу С. Л. Соболева, которая позволяет представить функцию, имеющую обобщенные производ-

ные до порядка l включительно, через эти производные. Точнее, если некий круг K_R радиуса R с центром в точке P полностью лежит в области Ω и введена осредняющая функция

$$\omega_R(Q) = \begin{cases} ce^{-\frac{R^2}{R^2-r^2}}, & r < R, \\ 0, & r \geq R, \quad r^2 = \zeta^2 + \eta^2 \end{cases} \quad (4.26)$$

такая, что

$$\int_{K_R} \omega_R(Q) dQ = 1, \quad (4.27)$$

любая функция $u(P)$ класса $W_2^l(\Omega)$ может быть представлена в виде

$$\begin{aligned} u(P) = & \sum_{\kappa=0}^{l-1} \sum_{l_1+l_2=\kappa} c_{l_1 l_2}^{(\kappa)}(P) \int_{\Omega} u(Q) \frac{\partial^{\kappa} \omega_R}{\partial x^{l_1} \partial y^{l_2}}(Q) dQ + \\ & + \sum_{l_1+l_2=l} \int A_{l_1 l_2}^{(l)}(P, Q) \frac{\partial^l u}{\partial x^{l_1} \partial y^{l_2}}(Q) dQ. \end{aligned} \quad (4.28)$$

При этом функции $c_{l_1 l_2}^{(\kappa)}(P)$ всегда ограничены в Ω . В частном случае $l=2$ имеем

$$\begin{aligned} u(P) = & \int_{K_R} u(Q) \omega_R(Q) dQ + c_1(P) \int_{K_R} u(Q) \frac{\partial \omega_R}{\partial x}(Q) dQ + \\ & + c_2(P) \int_{K_R} u(Q) \frac{\partial \omega_R}{\partial y}(Q) dQ + \sum_{l_1+l_2=2} \int_{K_R} A_{l_1 l_2}^{(2)}(P, Q) \frac{\partial^2 u}{\partial x^{l_1} \partial y^{l_2}}(Q) dQ, \end{aligned} \quad (4.29)$$

при этом

$$|A_{l_1 l_2}^{(2)}(P, Q)| \leq |a \ln r| + |b|, \quad r = \|P - Q\|_2. \quad (4.30)$$

Обозначим $\psi_N(P) = \varphi_*(P) - \sigma_N(P)$. Пусть N настолько велико, что $\varepsilon_N < \delta$. Рассмотрим круг радиуса $h = \varepsilon_N$ с центром в точке P , полностью лежащий в Ω , и сосредоточим в нем функцию $\omega_h(r)$. Представим $\psi_N(P)$ в соответствии с формулой (4.29),

$$\begin{aligned} \psi_N(P) = & \int_{K_h} \psi_N \omega_h dQ + c_1(P) \int_{K_h} \psi_N \frac{\partial \omega_h}{\partial x} dQ + \\ & + c_2(P) \int_{K_h} \psi_N \frac{\partial \omega_h}{\partial y} dQ + \sum_{l_1+l_2=2} \int_{K_h} A_{l_1 l_2}^{(2)}(P, Q) \frac{\partial^2 \psi_N}{\partial x^{l_1} \partial y^{l_2}}(Q) dQ. \end{aligned} \quad (4.31)$$

Применим неравенство Коши для оценки последней суммы в (4.31). Получим ($l_1 + l_2 = 2$) с учетом (4.30)

$$\left| \int_{K_h} A_{l_1 l_2}^{(2)}(P, Q) \frac{\partial^2 \psi_N}{\partial x^{l_1} \partial y^{l_2}}(Q) dQ \right| \leq \left\| \frac{\partial^2 \psi_N}{\partial x^{l_1} \partial y^{l_2}} \right\|_{L_2(\Omega)} \times \left[\int_0^{2\pi} d\varphi \int_0^h r (a \ln r + b)^2 dr \right]^{1/2}. \quad (4.32)$$

При достаточно малых h функция $|\ln r|$ может быть мажорирована функцией $r^{-\alpha}$ при сколь угодно малом $\alpha > 0$. Следовательно, с учетом (4.22) имеем

$$\left| \sum_{l_1 + l_2 = 2} \int_{K_h} A_{l_1 l_2}^{(2)}(P, Q) \frac{\partial^2 \psi_N}{\partial x^{l_1} \partial y^{l_2}}(Q) dQ \right| = o(1) \cdot h^{1-\alpha} = o(h^{1-\alpha}).$$

Далее, функции $\omega_h, \frac{\partial \omega_h}{\partial x}, \frac{\partial \omega_h}{\partial y}$ линейно-независимы. Представим функцию $\psi_N(P)$ в виде

$$\psi_N(P) = a_N + b_N x + c_N y + \varphi_N(P)$$

и выберем константы a_N, b_N, c_N так, чтобы первые три интеграла в разложении (4.31) для функции $\varphi_N(P)$ оказались равны 0. Это всегда возможно. Разложение (4.31) переписывается в виде

$$\psi_N(P) = a_N + b_N x + c_N y + o(h^{1-\alpha}). \quad (4.33)$$

В круге с центром в точке P радиуса h найдем три точки P_1, P_2, P_3 , входящие в интерполяционную сетку. Так как множество интерполяционных точек всюду плотно в Ω , мы можем обеспечить такое расположение точек, при котором площадь треугольника $P_1 P_2 P_3$ будет порядка h^2 , а его стороны — порядка h . Ясно, что

$$\psi_N(P_1) = \psi_N(P_2) = \psi_N(P_3) = 0.$$

Используем эти условия в (4.33). Очевидно, что линейная функция $a_N + b_N x + c_N y$ принимает в точках P_1, P_2 и P_3 значения порядка $o(h^{1-\alpha})$. Следовательно, она такова и внутри треугольника $P_1 P_2 P_3$, и во всем круге K_h .

Окончательно, используя тот факт, что из кругов типа K_h можно выделить конечное покрытие Ω_δ , имеем

$$\|\sigma_N - \varphi_*\|_{C(\Omega_\delta)} = o(h^{1-\alpha}). \quad (4.34)$$

Теорема доказана.

Пусть теперь в той же области Ω , звездной относительно некоторого шара, нам известны не значения функции $\varphi_* \in W_2^2(\Omega)$ на сгущающейся системе точек, а значения на сгущающейся системе спрямляемых гладких кривых Γ_i , объединение которых $\bigcup_{i=1}^{\infty} \Gamma_i$ всюду плотно в Ω . Введем в рассмотрение операторы A_i , сопоставляющие функции класса $W_2^2(\Omega)$ ее «след» на кривой Γ_i . Известно, что след функции класса $W_2^2(\Omega)$ на гладкой кривой является функцией, принадлежащей пространству Соболева — Слободецкого $W_2^{3/2}(\Gamma_i)$. Таким образом, мы определили систему операторов следов $A_i : W_2^2(\Omega) \rightarrow W_2^{3/2}(\Gamma_i)$. Утверждается, что это правильная система операторов. В самом деле, если имеет место сходимость некоторой последовательности функций φ_N к φ на кривых Γ_i по норме пространства $W_2^{3/2}(\Gamma_i)$, то по известным теоремам вложения [7] имеет место сходимость на Γ_i по норме $C(\Gamma_i)$, а значит, поточечная сходимость на множестве, всюду плотном в Ω . Совокупность функционалов, дающих значения функции на этом всюду плотном множестве точек, очевидно, всюду плотна в $W_2^2(\Omega)$ и образует требуемое множество K , на котором наблюдается слабая сходимость.

Итак, в этом случае применима теорема 4.1. Вывод оценок сходимости практически полностью совпадает с тем, что мы только что проделали. Как легко убедиться, для любой точки $P \in \Omega$ справедливо асимптотическое соотношение

$$|\sigma_N(P) - \varphi_*(P)| = o(h_N^{1-\alpha}), \quad (4.35)$$

где h_N — характерный параметр сгущения кривых в области Ω , определяемый так:

$$h_N = \sup_{x \in \Omega} \min_{i=1, \dots, N} \rho(x, \Gamma_i). \quad (4.36)$$

Здесь $\rho(x, \Gamma_i)$ — евклидово расстояние точки x до кривой Γ_i .

§ 4.4. СХОДИМОСТЬ СГЛАЖИВАЮЩИХ СПЛАЙНОВ К ИНТЕРПОЛЯЦИОННЫМ

Вернемся к общим формулировкам теории сплайнов. Пусть, как в гл. 1, X , Y и Z — три гильбертовых пространства, $T : X \rightarrow Y$, $A : X \rightarrow Z$ — линейные ограни-

ченные операторы, фиксирован элемент $z \in Z$ и множество $A^{-1}(z)$ не пусто. Через $\sigma \in X$ обозначим решение задачи интерполирования

$$\|T\sigma\|_Y^2 = \min_{A^{-1}(z)} \quad (4.37)$$

а через σ_α — решение задачи сглаживания:

$$\Phi_\alpha(\sigma_\alpha) = \alpha \|T\sigma_\alpha\|_Y^2 + \|A\sigma_\alpha\|_Z^2 = \min_X \quad (4.38)$$

Будем предполагать, что обе задачи однозначно разрешимы, т. е. (см. гл. 1) $N(A) \cap N(T) = \{\theta_X\}$, $R(T)$ и $R(A)$ замкнуты, ядро оператора T конечномерно. Изучим вопрос сходимости σ_α к σ при $\alpha \rightarrow 0$.

Лемма 1. Для любой последовательности x_n элементов ядра $N(T)$ из того, что $\|x_n\|_X \rightarrow \infty$, следует, что $\|Ax_n\|_Z \rightarrow \infty$.

Доказательство. Рассмотрим в $N(T)$ базисную систему $e_1, e_2, \dots, e_q$. Каждый из элементов e_i представим в виде $e_i = e_{i,1} + e_{i,2}$, где $e_{i,1} \in N(A)$, $e_{i,2} \in N(A)^\perp$. Тогда

$$x_n = \sum_{i=1}^q \lambda_i^{(n)} e_{i,1} + \sum_{i=1}^q \lambda_i^{(n)} e_{i,2}, \quad \sum_{i=1}^q \lambda_i^{n^2} \rightarrow \infty.$$

По условию $N(A) \cap N(T) = \{\theta_X\}$ элементы $Ae_{i,2}$ линейно-независимы, следовательно, норма вектора

$$Ax_n = \sum_{i=1}^q \lambda_i^{(n)} Ae_{i,2}$$

также стремится к бесконечности, что и требовалось.

Лемма 2. Последовательность сглаживающих сплайнов $\{\sigma_\alpha\}_{\alpha>0}$ ограничена в норме пространства X константой, не зависящей от α .

Доказательство. Очевидно, что $\Phi_\alpha(\sigma_\alpha) \leq \Phi_\alpha(\sigma)$ или подробнее

$$\|T\sigma_\alpha\|_Y^2 + \frac{1}{\alpha} \|A\sigma_\alpha - z\|_Z^2 \leq \|T\sigma\|_Y^2 + \frac{1}{\alpha} \|A\sigma - z\|_Z^2 = \|T\sigma\|_Y^2, \quad (4.39)$$

откуда имеем, разумеется,

$$\|T\sigma_\alpha\|_Y^2 \leq \|T\sigma\|_Y^2 \quad \forall \alpha > 0.$$

Представим σ_α в виде

$$\sigma_\alpha = \sigma_\alpha^1 + \sigma_\alpha^2,$$

где $\sigma_\alpha^1 \perp N(T)$, $\sigma_\alpha^2 \in N(T)$. Ясно, что оператор T осуществляет взаимно-однозначное соответствие $N(T)^\perp$ и $TX = R(T)$. Тогда по теореме об обратном операторе $\tilde{T}^{-1}$, являющийся обратным к сужению T на $N(T)^\perp$, непрерывен. Следовательно,

$$\|\sigma_\alpha^1\|_X \leq \|\tilde{T}^{-1}\| \cdot \|T\sigma_\alpha\|_Y \leq \|\tilde{T}^{-1}\| \cdot \|T\sigma\|_Y, \quad (4.40)$$

ограниченность последовательности $\{\sigma_\alpha^1\}$ показана.

Докажем теперь ограниченность $\{\sigma_\alpha^2\}$. Предположим противное, пусть существует подпоследовательность $\sigma_{\alpha'}^2$, такая, что $\|\sigma_{\alpha'}^2\|_X \rightarrow \infty$. Тогда по лемме 1 $\|A\sigma_{\alpha'}^2\|_Z \rightarrow \infty$. Конечно, это противоречит неравенству (4.39), в котором член $\frac{1}{\alpha'} \|A\sigma_{\alpha'} - z\|_Z^2$ в этом случае не может быть ограничен. Противоречие доказывает лемму.

Теорема 4.3. Сглаживающие сплайны σ_α слабо сходятся к интерполяционному сплайну σ при $\alpha \rightarrow 0$.

Доказательство. Пользуясь слабой компактностью ограниченных множеств в гильбертовых пространствах, выберем из последовательности $\{\sigma_\alpha\}$ слабо сходящуюся

$$\sigma_{\alpha'} \xrightarrow{c} \sigma_*. \quad (4.41)$$

Из условия (4.39) получаем

$$\|A\sigma_{\alpha'} - z\|_Z \rightarrow 0, \quad (4.42)$$

а, с другой стороны, из (4.41)

$$A\sigma_{\alpha'} \xrightarrow{c} A\sigma_*. \quad (4.43)$$

Следовательно,

$$A\sigma_* = z = A\sigma. \quad (4.44)$$

Так как элемент σ реализует минимум функционала $\|Tx\|_Y^2$ на $A^{-1}(z)$, то

$$\|T\sigma\|_Y^2 \leq \|T\sigma_*\|_Y^2. \quad (4.45)$$

Рассмотрим числовую последовательность

$$\mu_{\alpha'} = \|T\sigma_{\alpha'}\|_Y^2 + \frac{1}{\alpha'} \|A\sigma_{\alpha'} - z\|_Z^2. \quad (4.46)$$

Согласно (4.39), она ограничена. Выделим из нее подпоследовательность $\mu_{\alpha''}$, сходящуюся к некоторому пределу R . Ясно, что $R \leq \|T\sigma\|_Y^2$. Используем условие ортогональности для сглаживающих сплайнов (см. гл. 1) при $x = \sigma_*$. Получим

$$\begin{aligned}\mu_{\alpha''} &= (T\sigma_{\alpha''}, T\sigma_*)_Y + \frac{1}{\alpha} (A\sigma_{\alpha''} - z, A\sigma_* - z)_Z = \\ &= (T\sigma_{\alpha''}, T\sigma_*)_Y.\end{aligned}\quad (4.47)$$

Переходя в (4.47) к пределу при $\alpha'' \rightarrow 0$ с учетом слабой сходимости $T\sigma_{\alpha''}$ к $T\sigma_*$, имеем

$$\|T\sigma_*\|_Y^2 = R \leq \|T\sigma\|_Y^2. \quad (4.48)$$

Таким образом,

$$\begin{aligned}A\sigma_* &= A\sigma = z, \\ \|T\sigma_*\|_Y^2 &= \|T\sigma\|_Y^2.\end{aligned}\quad (4.49)$$

Так как интерполяционный сплайн σ единствен, $\sigma_* = \sigma$.

Следовательно, все предельные в слабой топологии точки последовательности $\{\sigma_\alpha\}$ совпадают с σ . Теорема доказана.

З а м е ч а н и е 1. Если оператор A имеет конечномерный образ, сглаживающие и интерполяционные сплайны лежат в одном и том же конечномерном пространстве и, конечно, имеет место сильная сходимость σ_α к σ .

З а м е ч а н и е 2. Из неравенства (4.39) можно немедленно заключить, что при $\alpha \rightarrow 0$ «певязка» при сглаживании $\|A\sigma_\alpha - z\|_Z$ стремится к нулю не медленнее $o(\sqrt{\alpha})$. Иначе величина $\frac{1}{\alpha} \|A\sigma_\alpha - z\|_Z^2$ была бы неограничена, что невозможно.

З а м е ч а н и е 3. Пусть A, T — некоторые матрицы, вообще говоря, прямоугольные и, быть может, вырожденные. Тогда интерполяционный сплайн σ — это T -нормальное решение системы

$$\begin{aligned}A\sigma &= z, \\ \|T\sigma\|_2^2 &= \min,\end{aligned}$$

которое, согласно теореме 4.3, может быть приближено вектором σ_α , доставляющим минимум функционалу

$$\alpha \|Tx\|_2^2 + \|Ax - z\|_2^2.$$

Условие минимума этого функционала может быть записано в виде «регуляризации» исходной системы

$$(\alpha T^*T + A^*A)\sigma_\alpha = A^*z.$$

§ 4.5. ВЫБОР ПАРАМЕТРА СГЛАЖИВАНИЯ

В рамках этого параграфа ограничимся сплайнами, относительно которых заданы линейно-независимые ограниченные функционалы $k_1, k_2, \dots, k_n$; размерность ядра оператора $T: X \rightarrow Y$ предполагается равной $q < \infty$.

При нахождении сглаживающего сплайна σ_α из условия минимума функционала

$$\alpha \|Tu\|_Y^2 + \sum_{i=1}^n [(k_i, u)_X - r_i]^2, \quad \alpha > 0, \quad (4.50)$$

возникает проблема разумного выбора параметра сглаживания α . По-видимому, самым популярным критерием выбора α является критерий невязки, который состоит в следующем: определим функцию

$$\varphi(\alpha) = \left(\sum_{i=1}^n [(k_i, \sigma_\alpha)_X - r_i]^2 \right)^{1/2} \quad (4.51)$$

и будем искать $\alpha > 0$ из условия

$$\varphi(\alpha) = \varepsilon, \quad (4.52)$$

где $\varepsilon > 0$ — допускаемый «уровень» уклонения сглаженных значений $(k_i, \sigma_\alpha)_X$ от заданных значений r_i . Уравнение (4.52) существенно нелинейное. Требуется определить интервал изменения ε , в котором (4.52) однозначно разрешимо, и предложить эффективный вычислительный алгоритм для нахождения α . С этой целью изучим функцию $\varphi(\alpha)$ более детально.

Согласно формулам (2.17), (2.18), вектор r_α сглаженных значений $(k_i, \sigma_\alpha)_X$ связан с исходным вектором соотношением

$$r_\alpha = r - \alpha H^* \lambda_\alpha,$$

где

$$(A + \alpha H H^*) \lambda_\alpha = H r.$$

Следовательно,

$$\varphi(\alpha) = \alpha \|H^* \lambda_\alpha\|_2 = \alpha \|H^* (A + \alpha H H^*)^{-1} H r\|_2. \quad (4.53)$$

Здесь $\|\cdot\|_2$ — евклидова метрика.

Теорема 4.4. Функция $\varphi(\alpha)$ представляет собой корень из дробно-рациональной функции вида

$$\varphi(\alpha) = \alpha \left[\sum_{i=1}^{n-q} \frac{a_i^2}{(1 + \alpha \lambda_i)^2} \right]^{1/2}, \quad \lambda_i > 0. \quad (4.54)$$

Доказательство. Согласно общей теории, развитой в гл. 2, $A = A^* > 0$. Следовательно, обобщенная проблема собственных значений

$$HH^*v = \lambda Av \quad (4.55)$$

разрешима, собственные векторы $v_1, v_2, \dots, v_{n-q}$ образуют базис, ортонормальный в скалярном произведении $[u, v]_A = (Au, v)_2$, собственные значения $\lambda_1, \lambda_2, \dots, \lambda_{n-q}$ положительны. Очевидно,

$$\varphi^2(\alpha) = \alpha^2 ((A + \alpha HH^*)^{-1} Hr, HH^*(A + \alpha HH^*)^{-1} Hr)_2.$$

Разложим вектор Hr по векторам $Av_1, Av_2, \dots, Av_{n-q}$,

$$Hr = \sum_{i=1}^{n-q} b_i Av_i. \quad (4.56)$$

Так как

$$(A + \alpha HH^*)^{-1} Av_i = (I + \alpha A^{-1} HH^*)^{-1} v_i = \frac{1}{1 + \alpha \lambda_i} v_i$$

получаем

$$\varphi^2(\alpha) = \alpha^2 \sum_{i,j=1}^{n-q} \frac{\lambda_j (v_i, Av_j)_2 b_i b_j}{(1 + \alpha \lambda_i)(1 + \alpha \lambda_j)} = \alpha^2 \sum_{i=1}^{n-q} \frac{\lambda_i b_i^2}{(1 + \alpha \lambda_i)^2},$$

что и требовалось, $a_i^2 = \lambda_i b_i^2 \geq 0$.

Положим $p = 1/\alpha$. Тогда

$$\psi^2(p) = \varphi^2(1/p) = \sum_{i=1}^{n-q} \frac{a_i^2}{(p + \lambda_i)^2}. \quad (4.57)$$

Путем вычислений легко устанавливается, что $1/\psi(p)$ на полуоси $[0, +\infty)$ — монотонно возрастающая от $1/\psi(0)$ до ∞ функция, которая, кроме того, выпукла вниз. Это означает, что уравнение

$$1/\psi(p) = 1/\varepsilon$$

имеет единственное решение при $0 < \varepsilon < \varepsilon_{\text{кр}} = \psi(0)$, причем решение может быть найдено по методу Ньютона с любым начальным приближением $p = p_0 > 0$.

Вычислим теперь величину $\psi(0) = \lim_{p \rightarrow 0} \psi(p)$. Из фор-

мулы (4.57) получим

$$\psi^2(0) = \sum_{i=1}^{n-q} \frac{b_i^2}{\lambda_i}.$$

Из разложения (4.56) следует

$$(Hr, v_i)_2 = \sum_{j=1}^{n-q} b_j (Av_j, v_i)_2 = b_i,$$

а из соотношения $HH^*v_i = \lambda_i Av_i$

$$(HH^*v_i, v_i)_2 = \lambda_i (Av_i, v_i)_2 = \lambda_i.$$

Таким образом,

$$\psi^2(0) = \sum_{i=1}^{n-q} \left[\frac{(r, H^*v_i)_2}{\|H^*v_i\|_2} \right]^2.$$

Векторы $e_i = H^*v_i / \|H^*v_i\|_2$ образуют ортонормированный базис в образе $R(H^*) = H^*E_{n-q}$ матрицы H^* . Действительно, $\|e_i\|_2 = 1$ и

$$(H^*v_i, H^*v_j)_2 = (HH^*v_i, v_j)_2 = \lambda_i (Av_i, v_j)_2 = 0, \quad i \neq j.$$

Окончательно величина $\psi(0)$ представляет собой евклидову длину проекции вектора r на $R(H^*)$, или, что то же самое, на плоскость $N(H)^\perp$, и для расчета $\varepsilon_{кр} = \psi(0)$ следует вычислить эту проекцию.

Пусть элементы $n_1, n_2, \dots, n_q$ образуют базис ядра оператора T . Тогда, как легко видеть, векторы

$$\bar{f}_i = [(k_1, n_i)_x, (k_2, n_i)_x, \dots, (k_n, n_i)_x], \quad i = \overline{1, q},$$

образуют базис ядра матрицы H . Действительно,

$$H\bar{f}_i = \left[\left(\sum_{j=1}^n \alpha_{kj} k_j, n_i \right)_x, \quad k = \overline{1, n-q} \right] = 0$$

(см. гл. 2, § 2.1). Длина проекции вектора r на $N(H)^\perp$ равна расстоянию от точки r до плоскости $N(H)$. Это приводит к условию

$$R^2(u) = \sum_{i=1}^n [(k_i, u)_x - r_i]^2 = \min_{u \in N(T)}.$$

Такая задача о наименьших квадратах эквивалентна системе линейных алгебраических уравнений порядка q

относительно коэффициентов разложения

$$u = \sum_{k=1}^q u_k n_k,$$

именно

$$F^* F \bar{u} = F^* r,$$

где $\bar{u} = [u_1, u_2, \dots, u_q]^T$, а матрица F составлена из векторов $\bar{f}_i$, как из столбцов. Решив эту систему, мы найдем минимальное значение величины $R(u)$, равное $\varepsilon_{кр}$.

Возможны следующие ситуации:

1. Пользователь задал $\varepsilon > \varepsilon_{кр}$ и пытается решить уравнение $\varphi(\alpha) = \varepsilon$. Конечно, решения α не существует. Заданный пользователем допустимый уровень невязки в процессе сглаживания настолько велик, что даже на элементах ядра оператора T возможны меньшие значения вариационного функционала сглаживания. Конечно, такую задачу сглаживания бессмысленно решать с помощью сплайнов — для ее решения, видимо, следует употребить другой метод.

2. $\varepsilon_{кр} = 0$. Это означает, что нашелся хотя бы один элемент ядра оператора T , на котором $\|Au - r\|_2 = 0$ и, следовательно, весь вариационный функционал сглаживания

$$\alpha \|Tu\|_Y^2 + \|Au - r\|_2^2 \equiv 0$$

независимо от α . Другими словами, вектор входных данных r вообще нельзя сгладить, $r_\alpha \equiv r$. Легко показать, что в этом случае

$$r = [(k_1, u)_x, (k_2, u)_x, \dots, (k_n, u)_x]^T,$$

где u — некоторый элемент ядра оператора T . Это следует из условия $Tr = 0$, которое влечет $\varphi(\alpha) \equiv 0$. Элемент u и будет решением задачи сглаживания при любом $\alpha > 0$.

Выведем теперь расчетные формулы для решения уравнения $\psi^{-1}(p) = \varepsilon^{-1}$ по методу Ньютона. Имеем ($\alpha = 1/p$):

$$p_{k+1} = p_k - \frac{\psi^{-1}(p_k) - \varepsilon^{-1}}{[\psi^{-1}(p_k)]'_p}, \quad (4.58)$$

где

$$[\psi^{-1}(p)]'_p = -\psi'_p(p) \psi^{-2}(p).$$

Согласно (4.53),

$$\varphi(\alpha) = \|r - r_\alpha\|_2 = \alpha \|H^* \lambda_\alpha\|_2, \quad (4.59)$$

где λ_α является решением системы

$$(A + \alpha HH^*)\lambda_\alpha = Hr. \quad (4.60)$$

Тогда

$$\begin{aligned} \varphi'(\alpha) = & (\alpha (H^*\lambda_\alpha, H^*\lambda_\alpha)^{1/2})'_\alpha = \|H^*\lambda_\alpha\|_2 + \\ & + \frac{\alpha (H^*\lambda'_\alpha, H^*\lambda_\alpha)_2}{\|H^*\lambda_\alpha\|_2}. \end{aligned}$$

При этом вектор λ'_α может быть получен путем дифференцирования обеих частей (4.60) по α :

$$(A + \alpha HH^*)\lambda'_\alpha = -HH^*\lambda_\alpha. \quad (4.61)$$

Таким образом, для расчета $\varphi(\alpha)$ и $\varphi'(\alpha)$ в фиксированной точке α необходимо решить системы (4.60) и (4.61) с одинаковой матрицей, но с различными правыми частями. Учитывая связь

$$\psi(p) = \varphi(1/p) = \varphi(\alpha), \quad \psi'_p(p) = \varphi'_\alpha(\alpha)(-\alpha^2),$$

запишем итерационный процесс (4.58) в терминах $\alpha_k = 1/p_k$:

$$\alpha_{k+1} = \alpha_k^2 \varphi'(\alpha_k) / (\alpha_k \varphi'(\alpha_k) - \varphi(\alpha_k) + \varphi^2(\alpha_k)/\varepsilon). \quad (4.62)$$

Отметим, что итерационный процесс (4.62) на практике сходится весьма быстро — 3—4 итерации уже приводят к условию $\varphi(\alpha) = \varepsilon \pm \varepsilon/10$. Положение дополнительно упрощается тем, что уравнение $\varphi(\alpha) = \varepsilon$ обычно не нужно решать очень точно, так как сам допустимый уровень невязки известен приближенно.

Глава 5

ПРИМЕНЕНИЕ ОСРЕДНЯЮЩИХ ОПЕРАТОРОВ ТИПА СВЕРТКИ ДЛЯ ОБРАБОТКИ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ

При решении многих практических задач приходится иметь дело с экспериментальными кривыми, от качественной обработки которых зависит успех решения задачи в целом. Как правило, экспериментальная кривая

представляет собой комбинацию некоторой гладкой составляющей, которая является «информативной», и кривой помех, порожденной самыми различными факторами. Задача обработки состоит в уменьшении помех и выделении информативной части, желательно без ее искажения.

В этой главе предлагается алгоритм обработки экспериментальной кривой, для которого исходными являются следующие предположения:

— известны, хотя бы приближенно, некоторые параметры информативной части кривой; могут быть известны степени многочленов, которые хорошо ее описывают, информативные частоты тригонометрических функций, показатели роста или убывания экспонент;

— кривая ошибок (помех) является высокочастотной по отношению к гладкой части кривой.

При выполнении этих условий гарантируется построение простейшего кусочно-постоянного локального ядра осредняющего оператора типа свертки, который, наряду со свойством подавления высокочастотных экспериментальных помех, оставляет неизменным заданный пользователем класс функций. Вычислительный процесс обработки очень прост и дешев.

§ 5.1. НЕКОТОРЫЕ СВОЙСТВА ОСРЕДНЯЮЩИХ ОПЕРАТОРОВ ТИПА СВЕРТКИ

На вещественной оси R^1 рассмотрим финитную функцию $K(t)$, $K(t) \in L_2(R^1)$. Через $LI_2(R^1)$ обозначим пространство функций на R^1 , которые квадратично интегрируемы на любом конечном интервале вещественной оси. Введем в рассмотрение оператор осреднения, порождаемый функцией $K(t)$ в пространстве $LI_2(R^1)$ по формуле

$$(K \hat{*} f)(x) = \int_{-\infty}^{\infty} K(y - x)f(y) dy. \quad (5.1)$$

Операция $\hat{*}$ осреднения функции $f(t)$ с помощью функции $K(t)$ очевидным образом связана с операцией свертки $*$ [4]:

$$K(t) \hat{*} f(t) = K(-t) * f(t). \quad (5.2)$$

Будем говорить, что осреднение, порожденное $K(t)$, со-

храняет функцию $f(t)$, если

$$K \hat{*} f = f. \quad (5.3)$$

Совершенно ясно, что если $K(t)$ сохраняет $f(t)$, то $K(t)$ сохраняет любой ее «сдвиг», т. е. функции вида $f(t - A)$ для всех $A \in R^1$. Легко показать, что при $f(t) \in C^1(R^1)$ условие $K \hat{*} f = f$ влечет выполнение условия $K \hat{*} f' = f'$, т. е. линейное пространство функций, сохраняемое оператором (5.1), инвариантно по отношению к дифференцированию.

В [4] дан полный ответ на вопрос, каково общее решение уравнения

$$\int_{-\infty}^{\infty} K(y - x) f(y) dy = f(x). \quad (5.4)$$

Именно, в пространстве функций экспоненциального роста общее решение уравнения (5.4) есть функция вида

$$f(t) = \sum_{i=1}^n \sum_{\kappa=0}^{m_i-1} A_{i\kappa} t^{\kappa} e^{z_{\kappa} t}, \quad (5.5)$$

где $n, m_1, m_2, \dots, m_n$ — некоторые натуральные числа, $z_1, z_2, \dots, z_n$ — некоторые комплексные числа, $A_{i\kappa}$ — неопределенные коэффициенты. Таким образом, оператор типа свертки может сохранять только функции вида (5.5), которые называют [4] квазиполиномами. Числа m_i, z_i, n определяются, разумеется, структурой функции $K(t)$. Справедлива

Теорема 5.1. Для того, чтобы осредняющий оператор (5.1) сохранял квазиполином $\varepsilon(t) = t^s e^{zt}$, необходимо и достаточно, чтобы преобразование Фурье $\hat{K}(\zeta)$ функции $K(t)$ удовлетворяло условиям

$$\hat{K}(-iz) = 1, \quad (5.6)$$

$$\frac{d^q}{d\zeta^q} \hat{K}(-iz) = 0, \quad q = 1, 2, \dots, s. \quad (5.7)$$

Доказательство. Мы хотим, чтобы выполнялось условие

$$(K \hat{*} \varepsilon)(x) = x^s e^{zx}. \quad (5.8)$$

Подробнее,

$$\begin{aligned}
 (\widehat{K * \varepsilon})(x) &= \int_{-\infty}^{\infty} K(t-x) t^s e^{zt} dt = \int_{-\infty}^{\infty} K(\mu) (\mu+x)^s e^{z(\mu+x)} d\mu = \\
 &= e^{zx} \int_{-\infty}^{\infty} K(\mu) (\mu+x)^s e^{z\mu} d\mu = e^{zx} \sum_{r=0}^s C_s^r x^{s-r} \int_{-\infty}^{\infty} K(\mu) \mu^r e^{z\mu} d\mu = \\
 &= e^{zx} \sum_{r=0}^s C_s^r x^{s-r} (i)^{-r} \int_{-\infty}^{\infty} K(\mu) (i\mu)^r e^{i(-iz)\mu} d\mu = \\
 &= e^{zx} \sum_{r=0}^s [(i)^{-r} C_s^r \widehat{K}^{(r)}(-iz)] x^{s-r}.
 \end{aligned}$$

Для того, чтобы последнее выражение равнялось $x^s e^{zx}$, необходимо и достаточно, чтобы

$$\begin{aligned}
 \widehat{K}(-iz) &= 1, \\
 \widehat{K}^{(r)}(-iz) &= 0, \quad r = 1, 2, \dots, s.
 \end{aligned}$$

Мы воспользовались здесь хорошо известным тождеством [9],

$$\widehat{K}^{(q)}(t) = \bigwedge (ix)^{(q)} K(x). \quad (5.9)$$

Следствие. Для того, чтобы осредняющий оператор (5.1) оставлял неизменными квазиполиномы $\varepsilon_{\kappa}(t) = t^{m_{\kappa}} e^{z_{\kappa} t}$, $\kappa = 1, 2, \dots, n$ (а значит, и их производные), необходимо и достаточно, чтобы функция $\widehat{K}(t) - 1$ имела в точках $-iz_1, -iz_2, \dots, -iz_n$ комплексной плоскости корни кратности $m_1 + 1, \dots, m_n + 1$.

Отметим, что функция $K(t)$ является вещественной, и, следовательно, условие $\widehat{K * \varepsilon} = \varepsilon$ влечет $\widehat{K * \operatorname{Re} \varepsilon} = \operatorname{Re} \varepsilon$, $\widehat{K * \operatorname{Im} \varepsilon} = \operatorname{Im} \varepsilon$. Это означает, что наряду с присутствием у функции $\widehat{K} - 1$ корня $(-iz)$ кратности s у нее есть корень в точке $(-i\bar{z})$ такой же кратности. Здесь $\bar{z}$ комплексно сопряжено z . Другими словами, квазиполиномы $t^s e^{zt}$ и $t^s e^{\bar{z}t}$ сохраняются одновременно.

§ 5.2. ОСРЕДНЯЮЩИЕ ФУНКЦИИ, ЛИНЕЙНО ЗАВИСЯЩИЕ ОТ КОНЕЧНОГО ЧИСЛА ПАРАМЕТРОВ

Как видно из теоремы 5.1, достаточно снабдить функцию $\widehat{K}(t) - 1$ нужными корнями нужной кратности, и мы получим оператор, сохраняющий заданную систе-

му квазиполиномов. Для этого надо ввести некоторое количество свободных параметров, изменяя которые, можно построить функцию $K(t)$ с заданными свойствами. Наиболее естественно рассмотреть линейную зависимость от свободных параметров.

Пусть задача состоит в построении осреднения, сохраняющего систему квазиполиномов вида

$$\varepsilon_k(t) = t^{m_k-1} e^{z_k t}, \quad k = 1, 2, \dots, n. \quad (5.10)$$

Следовательно, функция $\hat{K}(t) - 1$ должна обладать корнями $-iz_k, -i\bar{z}_k, k = 1, 2, \dots, n$, кратности $m_1, m_2, \dots, m_n$. Для определенности можно считать, что $\operatorname{Im} z_k \geq 0$ и среди точек z_k нет равных. Тогда общее количество p корней функции $\hat{K}(t) - 1$, включая кратные, вычисляется по формуле

$$p = \sum_{k=1}^n \omega(m_k, z_k), \quad \omega(s, z) = \begin{cases} s, & \operatorname{Im} z = 0, \\ 2s, & \operatorname{Im} z > 0. \end{cases} \quad (5.11)$$

Эту величину будем называть дифференциальной размерностью системы квазиполиномов $\varepsilon_k(t), k = 1, 2, \dots, n$, и обозначать так:

$$p = \operatorname{dimd}(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n). \quad (5.12)$$

Это понятие можно ввести иначе: рассмотрим квазимногочлены $\varepsilon_1(t), \dots, \varepsilon_n(t)$ и подвергнем их дифференцированию бесконечное число раз, затем образуем линейную оболочку вещественных и мнимых частей функций $\varepsilon_k^{(i)}(t), k = \overline{1, n}, i = 0, 1, 2, \dots$. Разумеется, это будет конечномерное пространство. Его размерность и есть дифференциальная размерность системы квазимногочленов. Например, дифференциальная размерность системы $1, e^{-0,5t}, \sin 2t$ равна 4.

Систему квазимногочленов будем называть дифференциально независимой (D -независимой), если

$$\operatorname{dimd}(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n) = \sum_{k=1}^n \operatorname{dimd}(\varepsilon_k). \quad (5.13)$$

Например, $1, e^t, \sin t$ — D -независима, $1 + 1 + 2 = 4$, а система $1, t^2, \sin t$ — D -зависима, $1 + 3 + 2 > 5$.

Итак, пусть p — дифференциальная размерность системы квазимногочленов, которые мы хотим сохранить в процессе осреднения. Будем искать $K(t) = K_p(t)$ в виде

$$K_p(t) = \sum_{l=1}^p \alpha_l \varphi_l(t), \quad (5.14)$$

где $\varphi_1, \varphi_2, \dots, \varphi_p$ — линейно-независимые вещественные функции на R^1 , $\alpha_1, \alpha_2, \dots, \alpha_p$ — вещественные параметры, которые предстоит выбрать. Так как

$$\widehat{K}(t) - 1 = \sum_{l=1}^p \alpha_l \widehat{\varphi}_l(t) - 1, \quad (5.15)$$

$$\widehat{K}^{(q)}(t) = \sum_{l=1}^p \alpha_l \widehat{\varphi}_l^{(q)}(t) = \sum_{l=1}^p \alpha_l \bigwedge (it)^q \varphi_l(t), \quad (5.16)$$

возникает следующая система относительно параметров α_l :

$$\sum_{l=1}^p \alpha_l \widehat{\varphi}_l(-iz_\kappa) = 1, \quad (5.17)$$

$$\sum_{l=1}^p \alpha_l \widehat{\varphi}_l(-i\bar{z}_\kappa) = 1,$$

$$\sum_{l=1}^p \alpha_l \widehat{\varphi}_l^{(q)}(-iz_\kappa) = 0, \quad q = 1, \dots, m_\kappa - 1, \quad \kappa = 1, \dots, n, \quad (5.18)$$

$$\sum_{l=1}^p \alpha_l \widehat{\varphi}_l^{(q)}(-i\bar{z}_\kappa) = 0, \quad q = 1, \dots, m_\kappa - 1, \quad \kappa = 1, \dots, n.$$

Это система с комплексной матрицей порядка $p \times p$. Разумеется, уравнения при $z_\kappa = \bar{z}_\kappa$ не повторяются дважды. Положим $z_\kappa = \eta_\kappa + i\xi_\kappa$. Тогда

$$\begin{aligned} \widehat{\varphi}_l^{(q)}(-iz_\kappa) &= \int_{-\infty}^{\infty} (it)^q \varphi_l(t) e^{(\eta_\kappa + i\xi_\kappa)t} dt = \\ &= (i)^q \left[\int_{-\infty}^{\infty} t^q e^{\eta_\kappa t} \cos \xi_\kappa t \varphi_l(t) dt + i \int_{-\infty}^{\infty} t^q e^{\eta_\kappa t} \sin \xi_\kappa t \varphi_l(t) dt \right]. \end{aligned}$$

Произведем с системой следующие эквивалентные преобразования: для всех κ , при которых $\text{Im } z_\kappa \neq 0$, найдем полусумму и полуразность пар уравнений (5.17) и (5.18). Сокращая при необходимости комплексный множитель, получим следующую эквивалентную систему:

$$\sum_{l=1}^p \alpha_l \left[\int_{-\infty}^{\infty} e^{\eta_\kappa t} \varphi_l(t) dt \right] = 1; \quad (5.19)$$

$$\sum_{l=1}^p \alpha_l \left[\int_{-\infty}^{\infty} t^q e^{\eta_\kappa t} \varphi_l(t) dt \right] = 0, \quad q = 1, \dots, m_\kappa - 1, \quad \kappa \in J_1,$$

$$\sum_{l=1}^p \alpha_l \left[\int_{-\infty}^{\infty} e^{\eta_{\kappa} t} \sin \zeta_{\kappa} t \varphi_l(t) dt \right] = 0, \quad (5.20)$$

$$\sum_{l=1}^p \alpha_l \left[\int_{-\infty}^{\infty} e^{\eta_{\kappa} t} \cos \zeta_{\kappa} t \varphi_l(t) dt \right] = 1,$$

$$\sum_{l=1}^p \alpha_l \left[\int_{-\infty}^{\infty} t^q e^{\eta_{\kappa} t} \cos \zeta_{\kappa} t \varphi_l(t) dt \right] = 0, \quad q=1, \dots, m_{\kappa}-1, \kappa \in J_2, \quad (5.21)$$

$$\sum_{l=1}^p \alpha_l \left[\int_{-\infty}^{\infty} t^q e^{\eta_{\kappa} t} \sin \zeta_{\kappa} t \varphi_l(t) dt \right] = 0, \quad q=1, \dots, m_{\kappa}-1, \kappa \in J_2.$$

Здесь J_1 — совокупность номеров, для которых $\operatorname{Im} z_{\kappa} = 0$, J_2 — оставшиеся номера, $J_1 \cup J_2 = \{1, 2, \dots, n\}$. Преобразованная система уже имеет вещественные коэффициенты, порядок матрицы по-прежнему $p \times p$.

Исследуем вопрос о существовании и единственности решения системы (5.19)–(5.21).

Теорема 5.2. Если преобразования Фурье $\widehat{\varphi}_{\kappa}$ функций φ_{κ} образуют в некотором компакте Ω комплексной плоскости T -систему Чебышева, система (5.19)–(5.21) однозначно разрешима для любых различных $z_1, z_2, \dots, z_n$, $\operatorname{Im} z_{\kappa} \geq 0$ при условии $-iz_{\kappa} \in \Omega$, $-i\bar{z}_{\kappa} \in \Omega$, $\kappa = 1, 2, \dots, n$.

Доказательство. Напомним, что функции $f_1, f_2, \dots, f_p$ образуют в компакте Ω T -систему Чебышева [8], если любая нетривиальная комбинация этих функ-

ций вида $\sum_{i=1}^p \alpha_i f_i$ имеет в Ω не более $(p-1)$ корней, с учетом кратных. Если система (5.19)–(5.21) имеет особенную матрицу, существует нетривиальная линейная комбинация $\sum_{l=1}^p \alpha_l^0 \widehat{\varphi}_l(t)$ с вещественными коэффициентами α_l^0 ,

которая имеет в Ω корни $-iz_{\kappa}$, $-i\bar{z}_{\kappa}$ кратности m_{κ} , т. е. p корней, что невозможно. Такой вывод можно сделать, вернувшись от вещественной системы (5.19)–(5.21) к комплексной (5.17), (5.18).

§ 5.3. КУСОЧНО-ПОСТОЯННЫЕ ОСРЕДНЯЮЩИЕ ФУНКЦИИ

В дальнейшем мы сосредоточим внимание на построении простейших локальных кусочно-постоянных ядер осредняющих операторов.

Рассмотрим вещественную ось с целочисленной решеткой $0, \pm 1, \pm 2, \dots$. Определим функцию

$$\varphi(t) = \begin{cases} 1, & -1/2 \leq t \leq 1/2, \\ 0 & \text{иначе} \end{cases} \quad (5.22)$$

и рассмотрим набор функций $\varphi_1, \varphi_2, \dots, \varphi_p$, вычисленных по формуле

$$\varphi_\kappa(t) = \varphi\left(t - \kappa + \frac{p+1}{2}\right), \quad \kappa = 1, 2, \dots, p. \quad (5.23)$$

Таким образом, каждая функция $\varphi_\kappa(t)$ испытывает разрывы в точках $\kappa - p/2 - 1, \kappa - p/2$. Любая линейная комбинация вида $\sum_{\kappa=1}^p \alpha_\kappa \varphi_\kappa(t)$, сосредоточенная в интервале $(-p/2, p/2)$, при четном p испытывает разрывы в целочисленных точках, при нечетных p — между целочисленными точками.

Найдем преобразование Фурье от функций φ_κ , $\kappa = 1, 2, \dots, p$. Ясно, что

$$\widehat{\varphi}(\zeta) = \int_{-1/2}^{1/2} e^{i\zeta t} dt = \frac{\sin(\zeta/2)}{\zeta/2}. \quad (5.24)$$

Учитывая очевидное равенство для преобразования Фурье сдвига функции

$$\widehat{f(t-a)} = e^{i\zeta a} \widehat{f(t)}, \quad (5.25)$$

получим

$$\widehat{\varphi}_\kappa(\zeta) = \frac{\sin(\zeta/2)}{\zeta/2} e^{-i\zeta \frac{p+1}{2}} e^{i\zeta \kappa}. \quad (5.26)$$

Задача состоит в определении таких компактов Ω на комплексной плоскости, в которых функции $\varphi_\kappa(\zeta)$ образуют T -системы Чебышева. Пусть $\alpha_1, \alpha_2, \dots, \alpha_p$ — нетривиальный набор параметров. Тогда

$$\sum_{\kappa=1}^p \alpha_\kappa \widehat{\varphi}_\kappa(\zeta) = \frac{\sin(\zeta/2)}{\zeta/2} e^{-i\zeta \frac{p+1}{2}} \sum_{\kappa=0}^{p-1} \alpha_{\kappa+1} (e^{i\zeta})^\kappa. \quad (5.27)$$

Функция $e^{-i\zeta \frac{p+1}{2}}$ не имеет корней в комплексной плоскости, функция $\frac{\sin(\zeta/2)}{\zeta/2}$ имеет корни в точках $\zeta = 2\pi n$, $n =$

$= \pm 1, \pm 2, \dots$. Оставшаяся часть $\sum_{\kappa=0}^{p-1} \alpha_{\kappa+1} (e^{i\zeta})^{\kappa}$ представляет собой многочлен степени $(p-1)$ от аргумента $e^{i\zeta}$. Прежде всего, исключим из предполагаемого компакта Ω точки $\zeta = 2\pi n$, $n = \pm 1, \pm 2, \dots$. Ясно, что существует не более $(p-1)$ чисел η_r таких, что

$$\sum_{\kappa=0}^{p-1} \alpha_{\kappa+1} \eta_r^{\kappa} = 0. \quad (5.28)$$

Теперь необходимо потребовать, чтобы для двух различных точек z' и z'' области Ω равенство $e^{iz'} = e^{iz''}$ было невозможно, т. е. $z' - z'' \neq 2\pi k$, $k = \pm 1, \pm 2, \dots$. Отсюда, в частности, следует

Теорема 5.3. Для любой системы квазимногочленов $\varepsilon_{\kappa}(t) = t^{n_{\kappa}} e^{z_{\kappa} t}$, $\kappa = 1, 2, \dots, n$, при условии, что числа $e^{z_1}, e^{z_2}, \dots, e^{z_n}$ различны, $\operatorname{Im} z_{\kappa} \geq 0$, существует единственная локальная кусочно-постоянная осредняющая функция $K_p(t)$, $p = \dim d(\varepsilon_1, \dots, \varepsilon_n)$, сохраняющая эти квазимногочлены.

§ 5.4. ОСРЕДНЕНИЯ, СОХРАНЯЮЩИЕ ТРИГОНОМЕТРИЧЕСКИЕ ФУНКЦИИ НАПЕРЕД ЗАДАННЫХ ЧАСТОТ

Построим осредняющий оператор, который сохраняет функции $\varepsilon_0(t) \equiv 1$, $\varepsilon_{\kappa}(t) = e^{i\zeta_{\kappa} t}$, где $\zeta_1, \zeta_2, \dots, \zeta_n > 0$ — различные ненулевые вещественные числа. Это означает, что будут сохраняться функции

$$1, \sin \zeta_{\kappa} t, \cos \zeta_{\kappa} t, \kappa = 1, 2, \dots, n. \quad (5.29)$$

Ясно, что дифференциальная размерность этой системы квазиполиномов

$$p = \dim d(\varepsilon_0, \varepsilon_1, \dots, \varepsilon_n) = 1 + 2n. \quad (5.30)$$

Значит, осредняющую функцию $K_p(t)$ следует искать в виде

$$K_{2n+1}(t) = \begin{cases} \alpha_{\kappa}, & t \in (\kappa - n - 3/2, \kappa - n - 1/2), \kappa = \overline{1, 2n+1}, \\ 0 & \text{иначе.} \end{cases}$$

Следуя общей теории, в точке $z_0 = 0$ и в точках $z_{\kappa} = \pm i(i\zeta_{\kappa}) = \pm \zeta_{\kappa}$ функция $\hat{K}_{2n+1}(z) - 1$ должна иметь кор-

ни кратности 1. Кроме того, для существования решения, по теореме 5.3, необходимо, чтобы

$$\zeta_k - \zeta_l \neq 2\pi m, \quad k \neq l, \quad m = 0, \pm 1, \pm 2, \dots$$

Итак, справедлива

Теорема 5.4. Для любых значений частот $\zeta_1, \zeta_2, \dots, \zeta_n, \zeta_k > 0$, таких, что $\zeta_k - \zeta_l \neq 2\pi m, \quad k \neq l, \quad m = 0, \pm 1, \pm 2, \dots$, существует единственная кусочно-постоянная осредняющая функция $K_{2n+1}(t)$, сохраняющая функции $1, \sin \zeta_k t, \quad k = \overline{1, n}$.

Следствие. Для любых различных $\zeta_k \in (0, 2\pi), \quad k = \overline{1, n}$, существует функция $K_{2n+1}(t)$, сохраняющая $1, \sin \zeta_k t, \quad k = \overline{1, n}$.

Выведем систему линейных алгебраических уравнений для определения коэффициентов $\alpha_1, \alpha_2, \dots, \alpha_{2n+1}$. Общая система (5.19)–(5.21) переписется в виде ($p = 2n + 1$):

$$\begin{aligned} \sum_{l=1}^p \alpha_l \int_{l-p/2-1}^{l-p/2} 1 dt &= 1, \\ \sum_{l=1}^p \alpha_l \int_{l-p/2-1}^{l-p/2} \cos \zeta_k t dt &= 1, \quad k = \overline{1, n}, \\ \sum_{l=1}^p \alpha_l \int_{l-p/2-1}^{l-p/2} \sin \zeta_k t dt &= 0, \quad k = \overline{1, n}. \end{aligned}$$

Производя интегрирование, получим

$$\sum_{l=1}^p \alpha_l = 1, \quad (5.31)$$

$$\sum_{l=1}^p \alpha_l \cos \zeta_k \left(l - \frac{p+1}{2} \right) = \frac{\zeta_k/2}{\sin(\zeta_k/2)}, \quad k = \overline{1, n}, \quad (5.32)$$

$$\sum_{l=1}^p \alpha_l \sin \zeta_k \left(l - \frac{p+1}{2} \right) = 0, \quad k = \overline{1, n}. \quad (5.33)$$

Система (5.31)–(5.33) всегда имеет «симметрическое» решение, т. е. $\alpha_l = \alpha_{p-l+1}$. В самом деле, при подстановке симметрического вектора уравнения (5.33) становятся тождествами. Поэтому порядок системы можно понизить

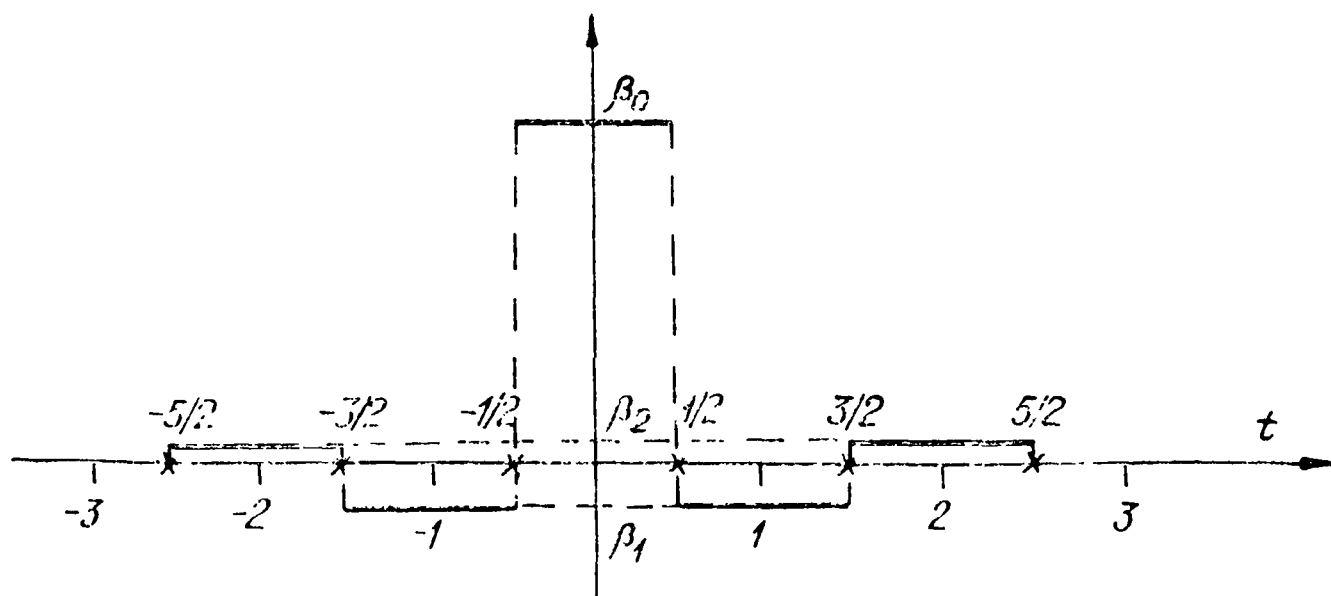


Рис. 6.

до $(n + 1)$. Пусть $\beta_l = \alpha_{n+l+1}$, $l = 0, 1, \dots, n$. Тогда с учетом симметрии

$$\beta_0 + 2 \sum_{l=1}^n \beta_l = 1, \quad (5.34)$$

$$\beta_0 + 2 \sum_{l=1}^n \beta_l \cos \zeta_\kappa l = \frac{\zeta_\kappa/2}{\sin(\zeta_\kappa/2)}, \quad \kappa = \overline{1, n}. \quad (5.35)$$

Пример. Построим осредняющую функцию $K_5(t)$ (рис. 6), сохраняющую $1, \sin t, \sin(t/2)$. Это соответствует $n = 2$, $\zeta_1 = 1$, $\zeta_2 = 0,5$. Система (5.34)–(5.35) примет вид

$$\begin{aligned} \beta_0 + 2\beta_1 + 2\beta_2 &= 1, \\ \beta_0 + (2 \cos 1)\beta_1 + (2 \cos 2)\beta_2 &= 0,5/\sin 0,5, \\ \beta_0 + (2 \cos 0,5)\beta_1 + (2 \cos 1)\beta_2 &= 0,25/\sin 0,25. \end{aligned}$$

Ее решение:

$$\begin{aligned} \beta_0 &= 1,116\,914\,535\,86, \\ \beta_1 &= -0,064\,119\,106\,391, \\ \beta_2 &= 0,005\,661\,838\,459, \\ \alpha_1 &= \beta_2, \quad \alpha_2 = \beta_1, \quad \alpha_3 = \beta_0, \quad \alpha_4 = \beta_1, \quad \alpha_5 = \beta_2. \end{aligned}$$

З а м е ч а н и е 1. Ограничение $\zeta_\kappa \neq 2\pi m$ означает, что невозможно сохранить гармонику, период которой кратен 1, т. е. шагу осреднения. Естественно, такая гармоника полностью аннулируется при интегрировании.

З а м е ч а н и е 2. Если задача состоит в выделении «длинноволнового» информативного фона и подавлении высокочастотной экспериментальной помехи, разумно выбирать сохраняемые частоты в полосе $[0, 2\pi - \delta)$. Конечно, эти частоты измерены по отношению к шагу осреднения, который мы принимали равным 1. Подробно вопрос о том, что происходит с промежуточными частотами, будет освещен в § 5.7.

З а м е ч а н и е 3. Вместо задачи на сохранение какой-либо частоты можно ставить задачу на ее «уничтожение»; для этого достаточно в правой части уравнения (5.35) вместо $\frac{\zeta_k/2}{\sin(\zeta_k/2)}$ поставить 0. Разумеется, можно усилить или ослабить заданную частоту.

§ 5.5. ОСРЕДНЕНИЯ, СОХРАНЯЮЩИЕ ЭКСПОНЕНЦИАЛЬНЫЕ ФУНКЦИИ ЗАДАННЫХ ПОКАЗАТЕЛЕЙ РОСТА ИЛИ УБЫВАНИЯ

Рассмотрим систему квазимногочленов $\varepsilon_0(t) \equiv 1$, $\varepsilon_k(t) = e^{\eta_k t}$, где $\eta_1, \eta_2, \dots, \eta_n$ — различные ненулевые вещественные числа. Дифференциальная размерность системы

$$p = \dim d(\varepsilon_0, \varepsilon_1, \dots, \varepsilon_n) = n + 1.$$

Осредняющую функцию будем искать в виде

$$K_{n+1}(t) = \begin{cases} \alpha_k, & t \in \left(k - \frac{n+1}{2} - 1, k - \frac{n+1}{2}\right), k = \overline{1, n+1} \\ 0 & \text{иначе.} \end{cases} \quad (5.36)$$

В соответствии с общей теорией функция $\hat{K}_{n+1}(z) - 1$ должна иметь корни кратности 1 в точках $z_k = \pm i\eta_k$ и в точке $z_0 = 0$. Так как для существования и единственности важны только ограничения на вещественные части z_k , а они все равны 0, справедлива.

Теорема 5.5. Для любых попарно различных $\eta_1, \dots, \eta_n$, $\eta_k \neq 0$, $k = \overline{1, n}$, существует единственная функция вида (5.36), сохраняющая квазиполиномы 1, $e^{\eta_k t}$, $k = \overline{1, n}$.

Система для определения α_l , $l = \overline{1, n+1}$, запишется так:

$$\sum_{l=1}^{n+1} \alpha_l \int_{l - \frac{n+1}{2} - 1}^{l - \frac{n+1}{2}} 1 dt = 1,$$

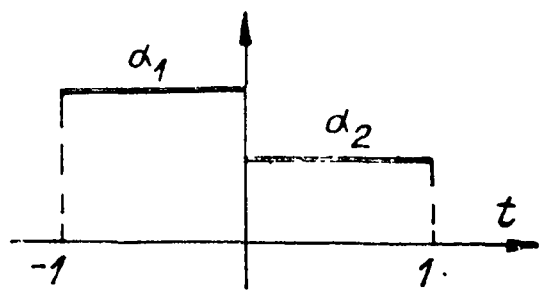


Рис. 7.

$$\sum_{l=1}^{n+1} \alpha_l \int_{l-\frac{n+1}{2}-1}^{l-\frac{n+1}{2}} e^{\eta_\kappa t} dt = 1, \quad \kappa = \overline{1, n},$$

или, после интегрирования,

$$\sum_{l=1}^{n+1} \alpha_l = 1, \quad (5.37)$$

$$\sum_{l=1}^{n+1} \alpha_l e^{\eta_\kappa l} = \frac{\eta_\kappa e^{\eta_\kappa \frac{n+1}{2}}}{1 - e^{-\eta_\kappa}}, \quad \kappa = \overline{1, n}. \quad (5.38)$$

Пример. Построим осреднение, сохраняющее 1 и e^t (рис. 7). Это соответствует $n = 1$, $\eta_1 = 1$. Получим следующую систему:

$$\begin{aligned} \alpha_1 + \alpha_2 &= 1, \\ e\alpha_1 + e^2\alpha_2 &= e/(1 - e^{-1}), \end{aligned}$$

ее решение:

$$\begin{aligned} \alpha_1 &= 0,661\,303\,112\,661\,5, \\ \alpha_2 &= 0,338\,696\,887\,338\,5. \end{aligned}$$

§ 5.6. ОСРЕДНЕНИЯ, СОХРАНЯЮЩИЕ ПОЛИНОМЫ ЗАДАННОЙ СТЕПЕНИ

Построим осредняющую функцию, сохраняющую полином t^n . Разумеется, такая функция будет сохранять любой полином степени n ,

$$p = \dim d(t^n) = n + 1.$$

Необходимо, чтобы осредняющая функция

$$K_{n+1}(t) = \begin{cases} \alpha_\kappa, & t \in \left(\kappa - \frac{n+1}{2} - 1, \kappa - \frac{n+1}{2}\right), \kappa = \overline{1, n+1}, \\ 0 & \text{иначе} \end{cases} \quad (5.39)$$

имела преобразование Фурье $\hat{K}_{n+1}(z)$ такое, что $\hat{K}_{n+1}(z) - 1$ в точке $z_0 = 0$ имело корень кратности $n + 1$. Это приводит к системе ($p = n + 1$):

$$\begin{aligned} \sum_{l=1}^{n+1} \alpha_l \int_{l-p/2-1}^{l-p/2} 1 dt &= 1, \\ \sum_{l=1}^{n+1} \alpha_l \int_{l-p/2-1}^{l-p/2} t^\kappa dt &= 0, \quad \kappa = 1, 2, \dots, n. \end{aligned}$$

После интегрирования получим систему

$$\sum_{l=1}^{n+1} a_{kl} \alpha_l = \begin{cases} 1, & k = 1, \\ 0, & k \neq 1, \end{cases} \quad (5.40)$$

где

$$a_{kl} = \left[\left(l - \frac{n+1}{2} \right)^k - \left(l - \frac{n+1}{2} - 1 \right)^k \right], \quad (5.41)$$

причем $k, l = 1, 2, \dots, n+1$. Справедлива

Теорема 5.6. Для любого натурального n существует единственная кусочно-постоянная функция $K_{n+1}(t)$ вида (5.39), сохраняющая многочлены степени $\leq n$.

Доказательство очевидным образом вытекает из теоремы 5.3.

Замечание. Как легко видеть, система (5.40) при любых n имеет «симметрическое» решение, т. е. $\alpha_l = \alpha_{n-l+2}$, $l = \overline{1, n+1}$. В самом деле, элементы матрицы (5.41) обладают очевидным свойством

$$a_{k, n-l+2} = (-1)^{k+1} a_{kl}, \quad k, l = \overline{1, n+1}.$$

Рассмотрим две ситуации.

1. n — нечетно. При подстановке симметрического решения в систему (5.40) уравнения при $k = 2, 4, \dots$ становятся тождествами. Если ввести переменные $\beta_l = \alpha_{(n+1)/2+l}$, $l = \overline{1, (n+1)/2}$, система (5.40) редуцируется к системе порядка $(n+1)/2$ вида

$$2 \sum_{l=1}^{\frac{n+1}{2}} a_{k, (n+1)/2+l} \beta_l = \begin{cases} 1, & k = 1, \\ 0, & k = 3, 5, \dots, n. \end{cases} \quad (5.42)$$

2. n — четно. При замене переменных $\beta_l = \alpha_{n/2+l}$, $l = \overline{1, (n+2)/2}$ система (5.40) трансформируется в систему

$$a_{k, (n+2)/2} \beta_1 + 2 \sum_{l=2}^{(n+2)/2} a_{k, (n+2)/2+l-1} \beta_l = \begin{cases} 1, & k = 1, \\ 0, & k = 3, 5, \dots, n+1. \end{cases} \quad (5.43)$$

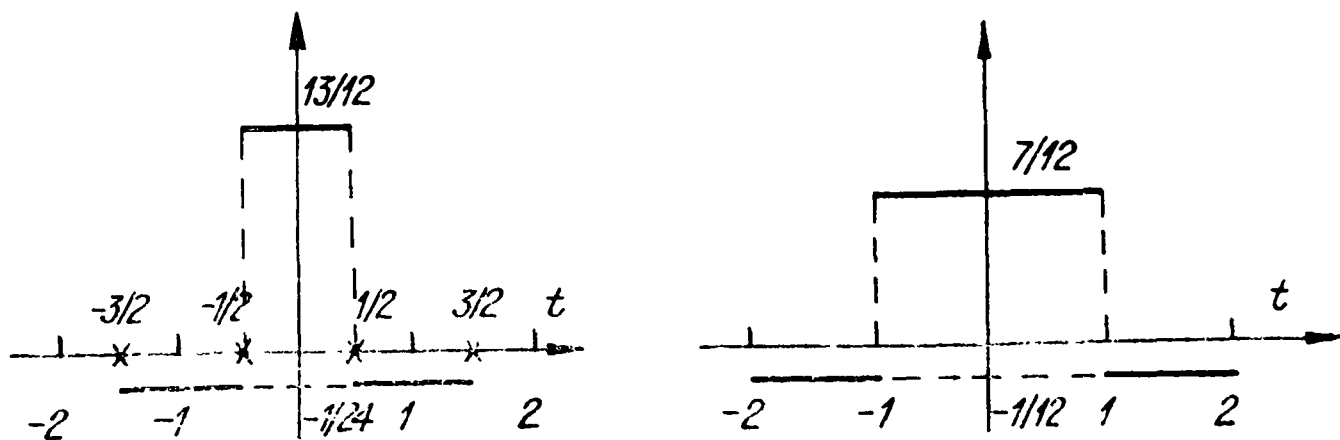


Рис. 8.

В заключение построим осредняющие функции, сохраняющие квадратичные ($n=2$) и кубические ($n=3$) многочлены (рис. 8). Системы (5.43) и (5.42) приобретают вид

$$\begin{aligned} & \begin{cases} \beta_1 + 2\beta_2 = 1, \\ \frac{1}{4}\beta_1 + \frac{26}{4}\beta_2 = 0, \end{cases} & \begin{cases} 2\beta_1 + 2\beta_2 = 1, \\ 2\beta_1 + 14\beta_2 = 0, \end{cases} \\ & \alpha_1 = \alpha_3 = \beta_2 = -\frac{1}{24}, \quad \alpha_1 = \alpha_4 = \beta_2 = -\frac{1}{12}, \\ & \alpha_2 = \beta_1 = \frac{13}{12}, \quad \alpha_2 = \alpha_3 = \beta_1 = \frac{7}{12}. \end{aligned}$$

§ 5.7. ЧТО ПРОИСХОДИТ, ЕСЛИ ПАРАМЕТРЫ ИНФОРМАТИВНОЙ ЧАСТИ КРИВОЙ ЗАДАНЫ НЕТОЧНО?

Как было отмечено, задавая входные параметры информативной части кривой, т. е. информативные частоты ζ_k , показатели роста или убывания экспонент η_k , степени многочленов n_k , можно эффективно построить операторы осреднения, сохраняющие нужные нам функции. Однако на практике все эти величины задаются приближенно, и важно уметь анализировать искажения, возникающие по этой причине.

Пусть $K_p(t)$ — кусочно-постоянная осредняющая функция, параметры которой $\alpha_1, \alpha_2, \dots, \alpha_p$ найдены, исходя из неких соображений, а $\varepsilon(t) = t^n e^{(\eta + i\xi)t} = t^n e^{zt}$ — квазимногочлен, параметры которого n, η, ξ произвольны. Посмотрим, что произойдет с ним при осреднении. Имеем ($z \neq 0$)

$$\begin{aligned} (K_p \hat{*} \varepsilon)(t) &= \sum_{\kappa=1}^p \alpha_{\kappa} \int_{t+\kappa-p/2-1}^{t+\kappa-p/2} \mu^n e^{z\mu} d\mu = \frac{n! e^{-pz/2}}{z^{n+1}} e^{zt} \times \\ &\times \sum_{k=1}^p \alpha_{\kappa} e^{\kappa z} \sum_{i=0}^n \frac{z^i}{i!} [(t + \kappa - p/2)^i - e^{-z} (t + \kappa - p/2 - 1)^i]. \end{aligned} \quad (5.44)$$

При $z = 0$ получим

$$\begin{aligned} (K_p \hat{*} \varepsilon)(t) &= \frac{1}{n+1} \sum_{\kappa=1}^p \alpha_{\kappa} [(t + \kappa - p/2)^{n+1} - \\ &- (t + \kappa - p/2 - 1)^{n+1}]. \end{aligned} \quad (5.45)$$

Таким образом, при осреднении квазиполном $\varepsilon(t)$ пре-

вращается снова в квазиполином $\varepsilon_{*}^{\wedge}(t)$ вида

$$\varepsilon_{*}^{\wedge}(t) = P_{n, z, \alpha}(t) e^{zt}, \quad (5.46)$$

где $P_{n, z, \alpha}(t)$ — многочлен снова степени n , линейно зависящий от параметров α_{κ} и нелинейно — от параметра z . Обратим внимание на то, что $P_{n, z, \alpha}(t)$ — вычислимая функция, и по ее поведению можно судить, что происходит с квазимногочленами, параметры которых не равны исходным.

Наиболее прост случай $n = 0$. Тогда функция e^{zt} становится собственной функцией оператора осреднения, причем соответствующее собственное число может быть немедленно найдено из формул (5.44) или (5.45):

$$K_p \hat{*} e^{zt} = A_{p, \alpha}(z) e^{zt}. \quad (5.47)$$

При этом

$$A_{p, \alpha}(z) = \frac{1}{z} e^{-pz/2} (1 - e^{-z}) \sum_{\kappa=1}^p \alpha_{\kappa} e^{\kappa z}, \quad (5.48)$$

$$A_{p, \alpha}(0) = \lim_{z \rightarrow 0} A_{p, \alpha}(z) = \sum_{\kappa=1}^p \alpha_{\kappa}. \quad (5.49)$$

Если z — чисто вещественное число, можно в соответствии с формулой (5.48) построить график функции $A_{p, \alpha}(z)$ и определить, что происходит с «соседними» по отношению к $e^{\eta_i t}$ экспонентами e^{zt} .

Если $z = i\zeta$ — чисто мнимое, то формулы (5.48), (5.49) переходят в

$$A_{p, \alpha}(i\zeta) = B_{p, \alpha}(\zeta) = \frac{\sin(\zeta/2)}{\zeta/2} \sum_{\kappa=1}^p \alpha_{\kappa} e^{i(\kappa - \frac{p+1}{2})\zeta}, \quad (5.50)$$

$$A_{p, \alpha}(0) = B_{p, \alpha}(0) = \sum_{\kappa=1}^p \alpha_{\kappa}. \quad (5.51)$$

Построив графики функций $\operatorname{Re} B_{p, \alpha}(\zeta)$ и $\operatorname{Im} B_{p, \alpha}(\zeta)$, мы сможем судить, что происходит с гармониками промежуточных частот по отношению к информативным $\zeta_1, \zeta_2, \dots, \zeta_n$, как меняются их амплитуды.

В качестве примеров приведем эти функции для осреднений, которые мы получили в § 5.4, 5.5.

Пример 1. Рассмотрим осредняющую функцию $K_5(t)$ (см. рис. 6), которая сохраняет частоты $\zeta_0 = 0, \zeta_1 = 1,$

Т а б л и ц а 5.7.1

ζ	B	ζ	B	ζ	B
1	1.000	8	0.963	15	0.422
2	1.000	9	0.926	16	0.322
3	1.000	10	0.872	17	0.250
4	1.000	11	0.800	18	0.176
5	0.999	12	0.713	19	0.111
6	0.995	13	0.618	20	0.053
7	0.985	14	0.519	21	0.000

Т а б л и ц а 5.7.2

z	A	z	A	z	A
1	1.000	8	0.966	15	1.095
2	0.985	9	0.974	16	1.129
3	0.974	10	0.985	17	1.167
4	0.966	11	1.000	18	1.209
5	0.961	12	1.018	19	1.257
6	0.960	13	1.040	20	1.310
7	0.961	14	1.065	21	1.368

$\zeta_2 = 0,5$. В силу того, что $K_5(t)$ — четная функция,

$$A_{5,\alpha}(i\zeta) = B_{5,\alpha}(\zeta) = \operatorname{Re} B_{5,\alpha}(\zeta)$$

и

$$B_{5,\alpha}(\zeta) = \frac{\sin(\zeta/2)}{\zeta/2} [\beta_0 + 2\beta_1 \cos \zeta + 2\beta_2 \cos 2\zeta], \quad (5.52)$$

$\beta_0, \beta_1, \beta_2$ определены в § 5.4. Значения функции (5.52) приведены в табл. 5.7.1 на отрезке $[0, 2\pi]$ с шагом $h = \pi/10$. Разумеется,

$$B_{5,\alpha}(0) = B_{5,\alpha}(1.0) = B_{5,\alpha}(0.5) = 1.$$

Пример 2. Рассмотрим теперь осредняющую функцию, сохраняющую 1, e^t , найденную в § 5.5. Это соответствует $\eta_0 = 0, \eta_1 = 1.0$. Для этого случая также

$$A_{2,\alpha}(z) = \operatorname{Re} A_{2,\alpha}(z), \quad -\infty < z < +\infty,$$

и

$$A_{2,\alpha}(z) = \frac{1}{z} e^{-z} (1 - e^{-z}) [\alpha_1 e^z + \alpha_2 e^{2z}], \quad (5.53)$$

α_1, α_2 определены в § 5.5. Значения функции $A_{2,\alpha}(z)$ приведены в табл. 5.7.2 на отрезке $[0, 2]$ с шагом $h = 0.1$. Конечно, $A_{2,\alpha}(0) = A_{2,\alpha}(1.0) = 1$.

§ 5.8. ВЫЧИСЛИТЕЛЬНЫЙ ПРОЦЕСС

Процесс осреднения зашумленного «сигнала» может быть осуществлен на устройствах разного типа. Это может быть фильтр, реализованный на уровне электронной схемы, аналоговое устройство и, разумеется, цифровая ЭВМ дискретного действия. Здесь мы рассмотрим только вычислительный процесс на устройствах последнего типа.

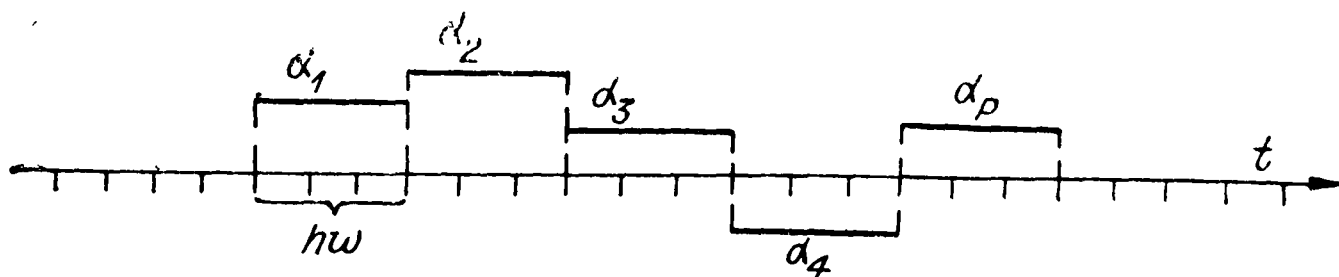


Рис. 9.

Современные средства позволяют перевести непрерывный сигнал в дискретный цифровой вид. Дискретизация связана с некоторой, обычно равномерной по времени, сеткой. Это позволяет сделать вычислительный процесс осреднения особенно быстрым. Программно-сеточную функцию удобно задавать в виде процедуры-функции целого аргумента, который является номером узла сетки. Такой тип оформления входной информации позволяет пользователю единым образом работать с различными источниками «сигнала» — датчиком сигнала или заранее рассчитанным массивом данных, который нужно обработать, не повредив информативный фон кривой и избавившись от высокочастотных шумов, вызванных, например, экспериментальными помехами.

Пусть на оси t (рис. 9) с шагом $h > 0$ с помощью процедуры-функции F целого аргумента j генерируется сеточная функция. При этом $F(j)$ должна «уметь» вычислять значения функции при изменении j в пределах некоторого интервала, границы которого в дальнейшем будут указаны.

Далее, пользователь должен указать «ширину» носителя осредняющей функции, т. е. количество w узлов сетки, которые укладываются в зону, где осредняющая функция постоянна.

Предположим, что мы хотим построить осредняющую функцию, сохраняющую квазиполиномы $t^{n_k} e^{\eta_k + i\zeta_k}$, $k = 1, 2, \dots, n$. Так как при теоретическом построении системы для определения параметров $\alpha_1, \alpha_2, \dots, \alpha_p$ (см. § 5.3—5.6) предполагалось, что $hw = 1$, следует изменить истинные входные параметры η_k и ζ_k , заданные в единицах пользователя, по формулам

$$\begin{aligned} \tilde{\eta}_k &= h \cdot w \cdot \eta_k, \\ \tilde{\zeta}_k &= h \cdot w \cdot \zeta_k, \quad k = \overline{1, n}. \end{aligned} \quad (5.54)$$

После этого можно пользоваться без всяких изменений ранее выведенными системами линейных уравнений и определять $\alpha_1, \alpha_2, \dots, \alpha_p$.

Пусть необходимо получить значения осредненной функции в узлах сетки, начиная с номера $N0$ и кончая номером NT . При интегрировании за основу можно принять простейшую формулу прямоугольников (вряд ли стоит очень точно интегрировать зашумленную кривую). Вычислим значение осреднения при $j = N0$. Возможны две ситуации:

1. pw — нечетное. Тогда, если $F_{CL}(\kappa)$ — осредненное значение в узле с номером κ , то

$$F_{CL}(N0) = \frac{1}{w} \sum_{i=1}^p \alpha_i \sum_{j=N0-w(p/2-i)+1/2}^{N0-w(p/2-i)-1/2} F(j). \quad (5.55)$$

2. pw — четное. Тогда

$$F_{CL}(N0) = \frac{1}{w} \sum_{i=1}^p \alpha_i \sum_{j=N0-w(p/2-i+1)}^{N0-w(p/2-i)-1} \frac{F(j) + F(j+1)}{2}. \quad (5.56)$$

Как только $F_{CL}(N0)$ вычислено, вступает в действие простое и очень экономичное с точки зрения вычислений рекуррентное соотношение

$$\begin{aligned} F_{CL}(\kappa + 1) = F_{CL}(\kappa) + \frac{1}{2w} \sum_{i=1}^p \alpha_i [& -F(\kappa - w(p/2 - i + 1)) - \\ & -F(\kappa + 1 - w(p/2 - i + 1)) + F(\kappa - w(p/2 - i)) + \\ & + F(\kappa + 1 - w(p/2 - i))], \end{aligned} \quad (5.57)$$

в ситуации 1

$$\begin{aligned} F_{CL}(\kappa + 1) = F_{CL}(\kappa) + \frac{1}{w} \sum_{i=1}^p \alpha_i [& -F(\kappa - w(p/2 - \\ & - i + 1) + 1/2) + F(\kappa - w(p/2 - i) + 1/2)]. \end{aligned} \quad (5.58)$$

Соотношения (5.57), (5.58) отражают тот факт, что при смещении осредняющей функции по оси на один шаг нет необходимости пересчитывать весь интеграл по формулам типа (5.55), (5.56), достаточно внести поправки, связанные со сдвигом вблизи точек разрыва осредняющей функции.

Во время тестовых расчетов ни разу не была зарегистрирована неустойчивость рекуррентных соотношений (5.57), (5.58), даже при $NT - N0 \cong 10^6$. Если же такие эффекты все-таки возникнут, следует просто разбить интересующий Вас интервал от $N0$ до NT на серию более коротких интервалов.

И последнее. Для того, чтобы вычислить значения осредняющей функции от номера $N0$ до номера NT необходимо, чтобы процедура-функция $F(j)$ генерировала значения сеточной функции в несколько более широком интервале, от $N0 - p - w/2 - 2$ до $NT + p + w/2 + 2$.

Описанный алгоритм очень быстр и прост в программировании.

Глава 6

ОПИСАНИЕ БИБЛИОТЕКИ ПРОГРАММ АППРОКСИМАЦИИ ФУНКЦИЙ *LIDA**)

Мы приступаем к описанию библиотеки программ, в которых реализованы на языке АЛГОЛ основные алгоритмы, изложенные в предыдущих главах.

Библиотека программ *LIDA* состоит из отдельных комплектов процедур. Каждый комплект предназначен для решения задач интерполяции или сглаживания определенного типа.

В библиотеку входят следующие комплекты:

- комплект *ODD* для расчета интерполяционных и сглаживающих сплайнов любой нечетной степени на отрезке;

- комплект *EVEN* для расчета сплайнов любой четной степени на отрезке по заданным локальным интегралам;

- комплект *GREEN* для интерполяции и сглаживания с помощью потенциалов в n -мерной области произвольной конфигурации с хаотическими сетками;

- комплект *AVERAGE* для построения и реализации осреднений, сохраняющих заданный квазиполиномиальный фон при обработке сильно зашумленных данных.

Описание каждого комплекта состоит из следующих основных частей:

- описание входящих в комплект процедур;
- описание отладочного теста;
- описание теста качественного характера;

*) Library on Data Approximation.

— замечания о работе с комплектом, требуемых ресурсах и т. д.;

— текст входящих в комплект программ.

Такая структура описаний позволит не только использовать описанные здесь программы, но и осуществить их тестирование при переходе на другой язык программирования.

Тексты программ приводятся в конце книги.

§ 6.1. КОМПЛЕКТ *ODD* ДЛЯ РАСЧЕТА ИНТЕРПОЛЯЦИОННЫХ И СГЛАЖИВАЮЩИХ СПЛАЙНОВ НЕЧЕТНОЙ СТЕПЕНИ НА ОТРЕЗКЕ

В § 2.3 детально описан алгоритм, который реализуется в комплекте *ODD*. Комплект *ODD* предназначен для построения интерполяционного или сглаживающего сплайна любой нечетной степени на отрезке по значениям в узлах произвольной неравномерной сетки. Сплайн выстраивается не только на отрезке между «крайними» узлами, но и за его пределами, что позволяет решать задачи экстраполяции. Комплект состоит из семи основных процедур, одной сервисной и вспомогательного массива, в котором расположены узлы и веса квадратурных формул Гаусса. Отметим, что комплект *ODD*, вообще говоря, ориентирован на решение серии задач сплайн-аппроксимации на одной сетке, но с разными входными данными. Это позволяет эффективно использовать его для решения многомерных задач на параллелепипедальных сетках (см. § 2.5).

6.1.1. Процедура *ASSH*

Процедура *ASSH* предназначена для формирования матрицы $\bar{H}$, см. § 2.3. Обращение к процедуре имеет вид

$$ASSH(N1, N2, X, P, H);$$

array $X[N1 : N2]$ — упорядоченный по возрастанию массив узлов сетки, наличие кратных узлов исключается.

integer $N1, N2$ — номера крайних узлов сетки, $N1 < N2$.

integer $P \geq 1$ — параметр, которым управляется степень сплайна $(2P - 1)$ и его гладкость C^{2P-2} , $P \leq 10$.

array $H[1 : N - P, 1 : P + 1]$ — массив-результат работы процедуры *ASSH*, $N = N2 - N1 + 1$ — количество узлов сетки.

При $P > N_2 - N_1$ сплайн-функция заданной степени не существует, на печать выдается текст 'КОЛИЧЕСТВО УЗЛОВ СЕТКИ НЕДОСТАТОЧНО ДЛЯ ПОСТРОЕНИЯ ИНТЕРПОЛЯНТА' и работа процедуры прекращается.

6.1.2. Процедура *ASSA*

Процедура *ASSA* предназначена для построения ленточной матрицы $\bar{A}$, см. § 2.3. Обращение к процедуре имеет вид

$ASSA(N_1, N_2, X, P, H, AL, A);$

Параметры N_1, N_2, X, P имеют прежний смысл, массив H выработан процедурой *ASSH*.

real $AL \geq 0$ — параметр сглаживания. Если решается интерполяционная задача, следует положить $AL = 0$. При $AL >_{10} - 17$ начинает работать режим сглаживания и вместо A формируется матрица $A + \alpha HH^*$, см. § 2.3.

array A — массив-результат работы процедуры *ASSH*. В зависимости от режима работы массив A имеет различные граничные пары. В режиме интерполирования при $AL = 0$ следует описать массив $A[1:N-P, 0:P-1]$, в режиме сглаживания при $AL >_{10} - 17$ массив имеет вид $A[1:N-P, 0:P]$. Обращаем внимание пользователя, что расширенный массив A , заимствованный из режима сглаживания, нельзя употреблять в режиме интерполирования.

Для обеспечения высокой скорости работы процедура *ASSA* использует при вычислении интегралов квадратуры Гаусса. Массив, в котором хранятся узлы и веса квадратурных формул, является внешним для всех процедур комплекта и имеет идентификатор *NW* (*nodes and weights*). Располагать его внутри *ASSA* нерентабельно, так как при каждом обращении к *ASSA* его нужно было бы заполнять, в то время как он совершенно неизменен. Поэтому все операции по его заполнению осуществляются один раз за пределами процедур комплекта. Массив *NW* устроен следующим образом:

array $NW[1:10, 1:10, 1:2];$

В нем расположены узлы и веса квадратур Гаусса с P узлами, начиная с $P = 1$ до $P = 10$. В ячейке массива $NW[i, j, 1]$ расположен j -й узел квадратуры с i узлами, $j \leq i$, в ячейке $NW[i, j, 2]$ — соответствующий вес. При необходимости этот массив можно расширить или, наобо-

рот, сократить. Например, если Вы в рамках своей программы используете только кубические сплайны ($P = 2$) и сплайны 5-й степени ($P = 3$), массив можно описать так:

array *NW*[2 : 3, 1 : 3, 1 : 2];

и оставить из всего набора присвоений только те, которые Вам необходимы.

6.1.3. Процедура *DECOMB*

Процедура *DECOMB* предназначена для получения разложения Холецкого матрицы A в режиме интерполяции или матрицы $A + \alpha HH^*$ в режиме сглаживания. Обращение к процедуре имеет вид

DECOMB (*N1*, *N2*, *P*, *AL*, *A*, *L*);

Параметры *N1*, *N2*, *P*, *AL* имеют прежний смысл.

array *A*[1 : $N - P$, 0 : κ] — массив, выработанный процедурой *ASSA*, причем $\kappa = P - 1$ в режиме интерполяции ($AL = 0$) и $\kappa = P$ в режиме сглаживания ($AL >_{10} - 17$).

array *L*[1 : $N - P$, 0 : κ] — массив-результат разложения A , $\kappa = P - 1$ при $AL = 0$ и $\kappa = P$ при $AL >_{10} - 17$.

Процедура *DECOMB* использует во время работы процедуру *CHOBANDDET* из комплекта Уилкинсона — Райнша [14] по линейной алгебре. Эта процедура осуществляет разложение Холецкого симметричной положительно определенной ленточной матрицы. Если матрица системы, которая подвергается разложению, заполнена неверно и оказалась вырожденной, на печать выдается текст 'ОШИБКА ВО ВХОДНЫХ ДАННЫХ ПРОЦЕДУРЫ *DECOMB*' и работа процедуры прекращается.

6.1.4. Процедура *SOLVE*

Процедура *SOLVE* предназначена для вычисления решения системы $A\lambda = Hr$ при $AL = 0$ или системы $(A + \alpha HH^*)\lambda_\alpha = Hr$ при $AL >_{10} - 17$. И в том, и в другом случае возможно решение систем со многими правыми частями, что обеспечивает возможность одновременного решения серии задач интерполяции или сглаживания. Обращение к процедуре имеет вид

SOLVE (*N1*, *N2*, *P*, *H*, *AL*, *R*, *L*, *F*, Φ);

Параметры $N1$, $N2$, P , AL имеют прежний смысл.

array $H[1:N-P, 1:P+1]$ — массив, выработанный процедурой *ASSH*.

array L — массив, выработанный процедурой *DECOMB*.

integer $R \geq 1$ — количество одновременно решаемых задач интерполяции или сглаживания.

array $F[N1:N2, 1:R]$ — массив значений сеточных функций в узлах, таких функций R штук.

array $\Phi[1:N-P, 1:R]$ — массив-результат работы процедуры *SOLVE*. По окончании работы *SOLVE* в массиве F находятся сглаженные значения, если сглаживание производилось ($AL >_{10} 17$).

Процедура *SOLVE* использует стандартную процедуру комплекта Уилкинсона — Райнша [14] *CHOBANDSOL*.

6.1.5. Процедура *REPRES*

Процедура *REPRES* предназначена для получения полиномиального представления серии сплайн-функций во всех ячейках сетки и за ее пределами (возможна экстраполяция). Именно в ячейке сетки (x_j, x_{j+1}) вычисляются коэффициенты разложения сплайна (безразлично, интерполяционного или сглаживающего) в виде

$$\sigma_j(x) = \sum_{\kappa=0}^{2P-1} A_j^{(\kappa)} \frac{(x - x_j)^\kappa}{\kappa!}, \quad j = N1, \dots, N2 - 1,$$

или, другими словами, вычисляются значения сплайна и его производных до порядка $(2P - 1)$ в узлах сетки. Кроме того, на интервале $(-\infty, x_{N1})$ сплайн-функция представляется в виде

$$\sigma_{N1-1}(x) = \sum_{\kappa=0}^{2P-1} A_{N1-1}^{(\kappa)} \frac{(x - x_{N1})^\kappa}{\kappa!},$$

а на интервале $(x_{N2}, +\infty)$ — в виде

$$\sigma_{N2}(x) = \sum_{\kappa=0}^{2P-1} A_{N2}^{(\kappa)} \frac{(x - x_{N2})^\kappa}{\kappa!}.$$

В действительности коэффициенты разложения сплайна за пределами интервала удовлетворяют условию (см. § 2.3)

$$A_{N1-1}^{(\kappa)} = A_{N2}^{(\kappa)} = 0, \quad \kappa = P, P+1, \dots, 2P-1.$$

Это означает, что экстраполяция реально осуществляется многочленом степени $P - 1$. Однако с целью сохранения единообразия массивов считается, что это полиномы степени $2P - 1$, нужные коэффициенты которых равны 0.

Обращение к процедуре имеет вид

REPRES ($N_1, N_2, P, X, F, R, \Phi, H, AA$);

Параметры N_1, N_2, P имеют прежний смысл.

array $X[N_1 : N_2]$ — массив интерполяционных точек.

array $F[N_1 : N_2, 1 : R]$ — массив значений сеточных функций (их R штук) в узлах сетки. Если происходило сглаживание, то массив выработан процедурой *SOLVE*.

integer $R \geq 1$ — количество одновременно решаемых задач интерполяции или сглаживания.

array $\Phi[1 : N - P, 1 : R]$ — массив, выработанный процедурой *SOLVE*.

array $H[1 : N - P, 1 : P + 1]$ — массив, выработанный процедурой *ASSH*.

array $AA[N_1 - 1 : N_2, 0 : 2 \times P - 1, 1 : R]$ — массив коэффициентов разложения сплайн-функций в ячейках сетки и за пределами интерполяционного интервала. Это результирующий массив процедуры *REPRES*.

В процессе работы процедуры *REPRES* необходимо решать серию систем линейных алгебраических уравнений порядка $(P - 1)$ с R правыми частями. Матрицы систем всегда неособенные, но не симметрические. Так как описываемая здесь библиотека программ реализована для АЛГОЛа — БЭСМ-6, мы использовали стандартную процедуру метода вращений *MVR*, обращение к которой имеет вид

MVR (a, b, n, m);

array $a[1 : n, 1 : n]$ — матрица системы (портится).

array $b[1 : m, 1 : n]$ — матрица, строками которой являются столбцы различных правых частей.

integer n — порядок системы.

integer m — число правых частей. Решения получаются на месте правых частей в соответствующих строках матрицы b .

При работе на ЭВМ с другой библиотекой стандартных программ процедуру *MVR* такого вида следует написать самому, конечно, с использованием имеющейся в Вашем распоряжении библиотеки стандартных процедур ЭВМ, на которой Вы работаете.

6.1.6. Вещественная процедура *FN*

Вещественная процедура-функция *FN* предназначена для вычисления значения сплайн-функции в любой точке вещественной оси. Обращение к ней имеет вид

$$Y := FN(N1, N2, P, X, AA, \kappa, X0);$$

Целые *N1*, *N2*, *P*, массивы *X*, *AA* имеют прежний смысл.

integer κ — номер сплайн-функции ($1 \leq \kappa \leq R$), значение которой Вы вычисляете.

real *X0* — координата точки на оси, в которой вычисляется значение сплайн-функции.

Разумеется, имея полное полиномиальное представление сплайна, полученное с помощью процедуры *REPRES*, можно выполнять различные операции: дифференцирование, интегрирование и т. д. Такие процедуры не включены в комплект и могут быть написаны самим пользователем. Мы ограничились только процедурой *FN*, как самой ходовой.

Наконец, комплект содержит одну сервисную процедуру, вычисляющую факториал. Расчет факториалов необходим в нескольких процедурах комплекта. Это целая процедура-функция, обращение

$$j := FAC(i);$$

приводит к $j = i!$.

6.1.7. Процедура выбора параметра сглаживания *BESPAR*

Процедура *BESPAR* предназначена для выбора параметра сглаживания α по критерию невязки (см. § 4.4) при построении сглаживающих сплайнов с помощью комплекта *ODD*. Если обозначить

$$\varphi(\alpha) = \left[1/(N2 - N1 + 1) \sum_{i=N1}^{N2} (\sigma_{\alpha}(x_i) - f_i)^2 \right]^{1/2}$$

и ввести уровень невязки $\varepsilon > 0$, то параметр α считается оптимальным при

$$|\varphi(\alpha) - \varepsilon| \leq \varepsilon/10.$$

Это разумное ограничение, так как уровень невязки ε обычно известен приближенно. Если, например, $\varepsilon = 0.1$, то пользователя, как правило, устраивает параметр α , при

котором

$$\varphi(\alpha) = 0.1 \pm 0.01.$$

Конечно, можно решить уравнение $\varphi(\alpha) = \varepsilon$ более точно, но при наличии ошибки в самом ε это приведет к бессмысленным затратам машинного времени.

Обращение к процедуре имеет вид

BESPAR (*P*, *N1*, *N2*, *X*, *F*, *EPS*, *AL*, *LAB*);

Параметры *P*, *N1*, *N2*, *X* имеют прежний смысл.

array *F* [*N1*:*N2*, 1:1] — массив сеточных значений сплайна, который подвергается сглаживанию; по окончании работы *BESPAR* в *F* находятся сглаженные значения сеточной функции.

real *FPS* > 0 — задаваемый пользователем уровень невязки.

real *AL* > 0 — перед началом работы *BESPAR* в *AL* находится начальное приближение параметра сглаживания; его следует выбирать достаточно большим ($AL \simeq 1$); после окончания работы *BESPAR* в *AL* находится оптимальный параметр.

label *LAB* — метка, по которой передается управление в случае, когда *EPS*, заданное пользователем, слишком велико. Это означает, что уравнение $\varphi(\alpha) = \varepsilon$ вовсе не имеет решения. Если это так, то бессмысленно использовать сплайн-функцию порядка $2P - 1$ для сглаживания, нужно построить полином степени $P - 1$, приближающий данные по методу наименьших квадратов. Как видно из описания, после работы *BESPAR* Вы получаете сглаженные сеточные значения. При этом величина *AL* уже не важна, так как в дальнейшем эти сглаженные значения можно обрабатывать комплектом *ODD* в режиме интерполяции. Однако в случае работы с серией однотипных таблиц данных можно, вычислив *AL* один раз на какой-нибудь одной таблице, использовать его для обработки всех остальных. Именно с этой целью величина *AL* сообщается пользователю.

6.1.8. Общие замечания по работе комплекта *ODD*

В этом пункте мы еще раз опишем последовательность действий при работе с комплектом *ODD*. Необходимо:

1. Расположить в программе процедуру *MVR*, взятую из стандартного обеспечения либо написанную пользователем.

2. Расположить в программе процедуры *CHOBANDDET* и *CHOBANDSOL* из комплекта программ по линейной алгебре Уилкинсона — Райнша [14].

3. Один раз обратиться к процедуре *ASSH*.

4. Один раз обратиться к процедуре *ASSA*.

5. Один раз обратиться к процедуре *DECOMB*.

Только после этого начинают использоваться значения в узлах сетки. Если Вы намерены решить одновременно R задач, то

6. Последовательность обращений

SOLVE (...);

REPRES (...);

приведет к построению полиномиального представления всех R интерполяционных и сглаживающих сплайнов.

6.1.9. Тест режима интерполяции

Пусть нам предстоит решить 2 задачи интерполяции сплайнами 5-й степени на одной сетке из 11 узлов. Пусть $N1 = 0$, $N2 = 10$, $R = 2$, $P = 3$, $AL = 0$. Сведем узлы сетки и значения в них искомых функций $\sigma_1(x)$ и $\sigma_2(x)$ в табл. 6.1.1.

Ясно, что $N = N2 - N1 + 1 = 11$. После обращения к *ASSH* вырабатывается массив $H[1:8, 1:4]$, см. табл. 6.1.2.

После обращения к *ASSA* вырабатывается массив $A[1:8, 0:2]$, см. табл. 6.1.3.

Разложение матрицы с помощью процедуры *DECOMB* приводит к массиву $L[1:8, 0:2]$, см. табл. 6.1.4.

Результатом работы процедуры *SOLVE* является массив $\Phi[1:8, 1:2]$, см. табл. 6.1.5.

Наконец, после обращения к процедуре *REPRES* вырабатывается массив $AA[-1:10, 0:5, 1:2]$, при этом $AA[-1:10, 0:5, 1]$ расположен в табл. 6.1.6, а $AA[-1:10, 0:5, 2]$ — в табл. 6.1.7.

С помощью процедуры *FN* подсчитаем значение сплайнов в точках: -0.5 (экстраполяция), 3.5 (интерполяция), 10.5 (экстраполяция), см. табл. 6.1.8.

6.1.10. Тест режима сглаживания

Сохраним неизменными сетку и сеточные значения сплайнов $\sigma_1(x)$ и $\sigma_2(x)$, см. табл. 6.1.1. Положим $AL = 1.0$. Обращение к процедуре *ASSH* приводит к заполнению массива $H[1:8, 1:4]$, см. табл. 6.1.2.

После обращения к *ASSA* вырабатывается массив $A[1:8, 0:3]$, см. табл. 6.1.9.

В результате разложения Холецкого матрицы A вырабатывается при обращении к *DECOMB* массив $L[1:8, 0:3]$, см. табл. 6.1.10.

Результатом работы процедуры *SOLVE* является массив $\Phi[1:8, 1:2]$, см. табл. 6.1.11.

Кроме того, после обращения к *SOLVE* исходный массив сеточных значений $F[0:10, 1:2]$ изменяется, в нем располагаются сглаженные сеточные значения сплайнов $\sigma_1(x)$ и $\sigma_2(x)$, см. табл. 6.1.12.

Наконец, после обращения к процедуре *REPRES* вырабатывается массив $AA[-1:10, 0:5, 1:2]$, при этом $AA[-1:10, 0:5, 1]$ расположен в табл. 6.1.13, а $AA[-1:10, 0:5, 2]$ — в табл. 6.1.14.

6.1.11. Тест процедуры *BESPAR*

Сохраним сетку, введенную в п. 6.1.8, и произведем выбор параметра AL для сплайна $\sigma_1(x)$, см. табл. 6.1.1. Положим $EPS = 0.05$, $P = 3$, $AL = 1$. Обращение

$BESPAR(3, 0, 10, X, F, 0.05, AL, LAB);$

приводит к оптимальному значению параметра

$$AL = 0.000\ 119\ 612\ 592\ 7.$$

§ 6.2. КОМПЛЕКТ *EVEN* ДЛЯ РАСЧЕТА ИНТЕРПОЛЯЦИОННЫХ И СГЛАЖИВАЮЩИХ СПЛАЙНОВ ЧЕТНОЙ СТЕПЕНИ НА ОТРЕЗКЕ ПО ЗАДАНЫМ ЛОКАЛЬНЫМ ИНТЕГРАЛАМ

Комплект *EVEN* по своей структуре практически идентичен комплекту *ODD*. Он также состоит из 7 основных процедур, сервисной процедуры *FAC*, вычисляющей факториал, и вспомогательного массива *NW*, в котором хранятся узлы и веса квадратурных формул Гаусса. Описания *FAC* и *NW* даны в п. 6.1.2, 6.1.6. Так же, как и комплект *ODD*, комплект *EVEN* использует стандартные процедуры *CHOBANDDET* и *CHOBANDSOL*, см. п. 6.1.3, 6.1.4, [14], и процедуру метода вращений *MVR*, см. п. 6.1.5.

Итак, комплект *EVEN* предназначен для построения интерполяционных либо сглаживающих сплайнов четной степени, относительно которых известны средние интег-

ральные значения по ячейкам сетки, см. § 2.4. Так же, как и комплект *ODD*, комплект *EVEN* предназначен для решения серии задач на одной и той же сетке, но с разными выходными данными.

6.2.1. Процедура *ASSH2*

Процедура *ASSH2* формирует ленточную матрицу $\bar{H}$, см. § 2.4. Обращение к ней имеет вид

$$ASSH2(N1, N2, X, P, H);$$

array $X[N1 : N2]$ — упорядоченный по возрастанию массив узлов сетки.

integer $N1 < N2$ — номера крайних узлов сетки.

integer $P > 0$ — параметр, которым управляется степень сплайна $2 \times P$ и гладкость C^{2P-1} .

array $H[1 : N - P - 1, 1 : P + 1]$ — массив-результат работы процедуры, $N = N2 - N1 + 1$ — количество узлов сетки. При $P > N2 - N1 - 1$ сплайн-функция заданной степени не существует, на печать выдается текст 'КОЛИЧЕСТВО УЗЛОВ СЕТКИ НЕДОСТАТОЧНО ДЛЯ ПОСТРОЕНИЯ ИНТЕРПОЛЯНТА ТАКОЙ СТЕПЕНИ' и работа процедуры прекращается.

6.2.2. Процедура *ASSA2*

Процедура *ASSA2* предназначена для построения ленточной матрицы $\bar{A}$, см. § 2.4. Обращение имеет вид

$$ASSA2(N1, N2, X, P, H, AL, A);$$

Параметры $N1, N2, X, P, H$ имеют прежний смысл, массив H выработан процедурой *ASSH2*.

real $AL \geq 0$ — параметр сглаживания. Если решается интерполяционная задача, следует положить $AL = 0$. При $AL \geq 10 - 17$ начинает работать режим сглаживания, вместо A формируется матрица $A + \alpha HH^*$, см. § 2.4.

array $A[1 : N - P - 1, 0 : P]$ — массив-результат работы процедуры *ASSA2*.

6.2.3. Процедура *DECOMB2*

DECOMB2 производит разложение Холецкого ленточной матрицы, полученной с помощью *ASSA2*. Обращение имеет вид

$$DECOMB2(N1, N2, P, A, L);$$

Параметры $N1$, $N2$, P , A имеют прежний смысл, массив A выработан процедурой $ASSA2$.

array $L[1 : N - P - 1, 0 : P]$ — массив-результат разложения. Если матрица системы заполнена неверно, то на печать выдается текст 'ОШИБКА ВО ВХОДНЫХ ДАННЫХ ПРОЦЕДУРЫ $DECOMB2$ '.

6.2.4. Процедура $SOLVE2$

Процедура $SOLVE2$ предназначена для вычисления решений системы со многими правыми частями. Обращение имеет вид

$$SOLVE2(N1, N2, P, H, AL, R, L, F, \Phi);$$

Параметры $N1$, $N2$, P , H , L , AL имеют прежний смысл. Массив H выработан процедурой $ASSH2$, массив L — процедурой $DECOMB2$.

integer $R \geq 1$ — количество одновременно решаемых задач.

array $F[N1 : N2 - 1, 1 : R]$ — массив интегральных средних интерполируемых или сглаживаемых функций (таких функций R штук).

array $\Phi[1 : N - P - 1, 1 : R]$ — массив-результат работы процедуры $SOLVE2$.

По окончании работы $SOLVE2$ в массиве F находятся сглаженные значения, если сглаживание проводилось ($AL \geq 10 - 17$).

6.2.5. Процедура $REPRES2$

Процедура $REPRES2$ предназначена для получения полиномиального представления серии сплайн-функций во всех ячейках сетки и за ее пределами (возможна экстраполяция). Точнее, в ячейке сетки $[x_m, x_{m+1}]$ вычисляются коэффициенты представления сплайна в виде

$$\sigma_m(x) = \sum_{j=0}^{2P} A_m^{(j)} \frac{(x - x_m)^j}{j!}, \quad m = N1, \dots, N2 - 1,$$

или, другими словами, значения сплайна и его производных до порядка $2P$ в узлах сетки. Кроме того, в интервале $(-\infty, x_{N1})$ сплайн-функция представляется в виде

$$\sigma_{N1-1}(x) = \sum_{j=0}^{2P} A_{N1-1}^{(j)} \frac{(x - x_{N1})^j}{j!},$$

а на интервале $(x_{N_2}, +\infty)$ — в виде

$$\sigma_{N_2}(x) = \sum_{j=0}^{2P} A_{N_2}^{(j)} \frac{(x - x_{N_2})^j}{j!}.$$

Обращение к процедуре имеет вид

REPRES2(*N1*, *N2*, *P*, *X*, *F*, *R*, *Ф*, *H*, *AA*);

Параметры *N1*, *N2*, *P*, *X*, *F*, *R*, *Ф*, *H* имеют прежний смысл, массив *Ф* выработан процедурой *SOLVE2*.

array *AA*[*N1* — 1 : *N2*, 0 : 2 × *P*, 1 : *R*] — массив коэффициентов разложений сплайн-функции в ячейках сетки и за ее пределами, результат работы *REPRES2*.

6.2.6. Вещественная процедура-функция *FN2*

Вещественная процедура *FN2* предназначена для вычисления значения сплайна в любой точке *X0* вещественной оси. Обращение к ней имеет вид

Y := *FN2*(*N1*, *N2*, *P*, *X*, *AA*, *κ*, *X0*);

Параметры *N1*, *N2*, *P*, *X*, *AA* имеют прежний смысл.

integer *κ* — номер сплайн-функции ($1 \leq \kappa \leq R$), значение которой предполагается вычислять.

real *X0* — координата точки на оси *x*, в которой вычисляется сплайн-функция.

6.2.7. Процедура выбора параметра сглаживания *BESPAR2*

Процедура *BESPAR2* в комплекте *EVEN* аналогична процедуре *BESPAR* в комплекте *ODD*. Обращение

BESPAR(*P*, *N1*, *N2*, *X*, *F*, *EPS*, *AL*, *LAB*);

приводит к выбору оптимального в соответствии с критерием невязок параметра сглаживания *AL*.

Параметры *P*, *N1*, *N2*, *X* имеют прежний смысл.

array *F*[*N1* : *N2* — 1, 1 : 1] — массив средних интегральных значений по ячейкам сетки; по окончании работы *BESPAR2* в нем хранятся сглаженные значения, с которыми можно работать в режиме интерполяции.

real *EPS* > 0 — величина, по которой останавливается итерационный процесс выбора *AL* > 0; точнее, процесс

заканчивается, если

$$\left| \sqrt{\frac{1}{N_2 - N_1 - 2} \sum_{i=N_1}^{N_2-1} \left[\frac{1}{x_{i+1} - x_i} \int_{x_i}^{x_{i+1}} \sigma_\alpha(x) dx - F[i, 1] \right]^2} - EPS \right| \leq EPS/10.$$

real $AL > 0$ — параметр сглаживания; перед началом работы в AL хранится начальное приближение к α ; предпочтительно $AL = 1$; после окончания работы в AL — оптимальное значение параметра.

label LAB — метка, по которой передается управление, если EPS слишком велико и оптимального параметра не существует; это означает, что задачу следует решать по методу наименьших квадратов полиномами степени $(P - 1)$.

6.2.8. Общие замечания по работе с комплектом *EVEN*

Еще раз опишем последовательность действий при работе с комплектом *EVEN*.

1. Следует расположить в начале программы текст процедуры *MVR*, предназначенной для решения системы линейных алгебраических уравнений с несимметрической плотной матрицей со многими правыми частями; предпочтительно использовать метод вращений, см. § 6.1.

2. Далее, расположить процедуры *CHOBANDDET* и *CHOBANDSOL* для разложения Холецкого и решения систем с ленточной симметрической положительно определенной матрицей, см. [14].

3. Расположить следом процедуры комплекта *EVEN*.

4. Заполнить необходимые элементы сервисного массива NW , см. § 6.1.

5. Следует один раз обратиться к процедуре *ASSH2*.

6. Один раз обратиться к *ASSA2*.

7. Один раз обратиться к *DECOMB2*.

Только после этого начинают использоваться средние интегральные значения.

8. Последовательность обращений

SOLVE2(...),
REPRES2(...)

приводит к построению полиномиального представления всех R интерполяционных или сглаживающих сплайнов, и задача решена.

6.2.9. Тест режима интерполяции

Положим, как в § 6.1, $N1 = 0$, $N2 = 10$, $P = 2$, $R = 2$, $AL = 0$. Будем решать задачу одновременного построения двух сплайнов 4-й степени. Координаты узловых точек и значения средних интегралов по ячейкам сетки приведены в табл. 6.2.1.

После обращения к *ASSH2* вырабатывается массив $A[1:8, 1:3]$, см. табл. 6.2.2.

Далее, при обращении к *ASSA2* вырабатывается массив $A[1:8, 0:2]$, см. табл. 6.2.3.

Результат разложения матрицы с помощью *DECOMB2* есть массив $L[1:8, 0:2]$, см. табл. 6.2.4.

Решение системы с двумя ($R = 2$) правыми частями приводит к массиву $\Phi[1:8, 1:2]$, см. табл. 6.2.5.

С помощью *REPRES2* получаем массив $AA[-1:10, 0:4, 1:2]$, массив $AA[-1:10, 0:4, 1]$ изобразим в табл. 6.2.6, массив $AA[-1:10, 0:4, 2]$ — в табл. 6.2.7.

С помощью *FN2* подсчитаем значения сплайнов $\sigma_1(x)$ и $\sigma_2(x)$ в точках $x = -0.5, 3.5, 10.5$, см. табл. 6.2.8

6.2.10. Тест режима сглаживания

Сохраним входные данные задачи, см. табл. 6.2.1, и положим $AL = 1$. Массив H остается неизменным, см. табл. 6.2.2. После обращения к *ASSA2* получим $A[1:8, 0:2]$, см. табл. 6.2.9.

После обращения к *DECOMB2* получим $\Phi[1:8, 0:2]$, см. табл. 6.2.10.

После обращения к *SOLVE2* получим $\Phi[1:8, 1:2]$, см. табл. 6.2.11.

Обращение к *REPRES2* приводит к массиву $AA[-1:10, 0:4, 1]$, см. табл. 6.2.12, и массиву $AA[-1:10, 0:4, 2]$, см. табл. 6.2.13.

Наконец, с помощью *FN2* подсчитаем значения сплайнов σ_1 и σ_2 в точках $x = -0.5, 3.5, 10.5$, см. табл. 6.2.14.

6.2.11. Тест процедуры *BESPAR2*

Сохраним сетку, введенную в п. 6.2.8, и выберем параметр AL для сплайна $\sigma_1(x)$, см. табл. 6.2.1. Положим $EPS = 0.05$, $P = 2$, $AL = 1$. Обращение

BESPAR2(2, 0, 10, X , F , 0.05, AL , LAB);

приводит к оптимальному значению параметра

$$AL = 0.007\,069\,347\,834.$$

§ 6.3. КОМПЛЕКТ *GREEN* ДЛЯ СПЛАЙН-АППРОКСИМАЦИИ В МНОГОМЕРНЫХ ОБЛАСТЯХ ПРОИЗВОЛЬНОЙ КОНФИГУРАЦИИ С ХАОТИЧЕСКИМИ СЕТКАМИ

В комплекте *GREEN* реализованы алгоритмы построения решений задач сплайн-интерполяции и сглаживания с помощью функции Грина полигармонического оператора, описанные в § 2.6, 2.7. Исходная постановка задачи такова: в n -мерной области произвольным образом расположены интерполяционные точки; возможны две ситуации: первая — известны только значения функции в этих точках, вторая — известны еще и значения первых и вторых производных по заданным направлениям, т. е. существует возможность учитывать наклоны и кривизны в заданных точках.

При решении задачи в обеих ситуациях необходимо решать систему линейных алгебраических уравнений с плотной матрицей, которая всегда симметрична, но не положительно определена. Этот этап является самым дорогостоящим, как с точки зрения расхода машинного времени, так и с точки зрения расхода оперативной памяти ЭВМ. Существует много практических задач, в которых положение интерполяционных точек (а значит, матрица системы) остается неизменным, меняются только значения функции или производных от варианта к варианту (т. е. правая часть системы). Поэтому рентабельно найти обратную матрицу, а затем решать систему с любой правой частью. Это обстоятельство и определило структуру комплекта *GREEN*.

В комплекте *GREEN* допускается (при определенных ограничениях) построение сплайн-функции любой степени гладкости. Точнее, гладкость регулируется параметром m , входящим в вариационный функционал (см. § 2.7)

$$\sum_{|\alpha|=m} \int_{\Omega} (D^{\alpha} u)^2 d\Omega,$$

размерностью n области Ω и числом k , зависящим от характера входной информации. При этом $k=0$, если заданы только значения функции в узлах, $k=1$, если среди входных данных есть значения первых производных, и $k=2$, если встречаются значения вторых производных. Три числа (m, n, k) определяют гладкость сплайн-функции, а именно, сплайн принадлежит классу

$C^{2m-n-k-1}(\Omega)$, и разрывы производных порядка выше $2m - n - k - 1$ возможны только в интерполяционных узлах. Например, в плоской области Ω ($n=2$) при $m=2$ и $k=0$ сплайн обладает гладкостью $C^1(\Omega)$. Прежде чем приступить к описанию комплекта *GREEN*, мы опишем параметры процедуры обращения матрицы. Она должна удовлетворять следующим требованиям: обращать произвольную симметрическую невырожденную матрицу и реагировать на возможную вырожденность. Подходящая процедура в стандартном обеспечении АЛГОЛа — БЭСМ-6 — это *INVERT2*. Она работает на основе метода Гаусса с выбором главного элемента по всей матрице и располагает обратной матрицу на месте исходной (это важно при обращении матрицы большого порядка). Обращение к ней имеет вид

INVERT2(*A*, *N*, *EPS*, *ALARM*);

array *A*[1 : *N*, 1 : *N*] — обрабатываемая матрица, A^{-1} будет помещена на место *A*.

integer *N* — порядок матрицы.

label *ALARM* — метка, на которую будет передано управление, если в процессе обращения ведущий элемент окажется меньше *EPS*. Процедура *INVERT2* может быть создана на основе стандартного обеспечения ЭВМ, на которой Вы работаете.

Цепочка процедур *FORMH*, *SOL*, *VAL* предназначена для построения интерполяционного или сглаживающего сплайна по произвольной системе узловых точек в *n*-мерной области, в которых заданы только значения функции.

6.3.1. Процедура *FORMH*

Процедура *FORMH* предназначена для формирования и обращения матрицы *H* порядка $N + ii$, где *N* — количество интерполяционных точек, $ii = (n + m - 1)! / (m - 1)! / n!$. Обращение к процедуре имеет вид

FORMH(*N*, *m*, *n*, *R0*, *ii*, *A0*, *A*, *H*, *J1*, *MA*, *MI*, *ARM*);

integer *N* — количество узлов хаотической сетки.

integer *m* — параметр вариационного функционала.

integer *n* — размерность области.

real $R0 \geq 0$ — параметр сглаживания, в режиме интерполяции следует положить $R0 = 0$. В режиме сгла-

живания минимизируется функционал

$$\Phi(u) = \int_{\Omega} \sum_{|\alpha|=m} (D^{\alpha}u)^2 d\Omega + R0 \sum_{i=1}^N [u(P_i) - f_i]^2.$$

integer ii — количество слагаемых в полиноме степени $m-1$ от n переменных, $ii = (n+m-1)!/(m-1)!/n!$ подсчитывается пользователем.

array $A0[1:N, 1:n]$ — массив координат интерполяционных точек, $A0[i, j]$ — это j -я координата i -й точки.

real MA, MI — величины, равные соответственно $MI = \min A0[i, j] - 10 - 5$, $MA = \max A0[i, j] - MI$, вырабатываются внутри *FORMH*.

array $A[1:N, 1:n]$ — массив координат интерполяционных точек, полученный при замене переменных $t_i = \frac{x_i - MI}{MA}$, $i = 1, 2, \dots, n$. Эта замена переменных переводит интерполяционные узлы в единичный куб. Массив вырабатывается *FORMH*.

array $H[1:N+ii, 1:N+ii]$ — массив-результат, содержащий обращенную матрицу.

integer array $J1[1:n, 1:ii]$ — массив, вырабатываемый *FORMH*, смысл которого мы поясним при описании процедуры *SOL*. Параметры должны удовлетворять неравенствам, необходимым для существования и единственности сплайна:

$$2m > n, \quad N \geq ii.$$

Если эти неравенства не выполнены или среди интерполяционных точек встречаются совпадающие, на печать выдается текст 'ВЫРОЖДЕНА' и управление передается на метку *ARM*.

6.3.2. Процедура *SOL*

Процедура *SOL* использует массив H , выработанный *FORMH*, и предназначена для расчета коэффициентов аналитического представления сплайна, именно: вырабатывает массив $B[1:N+ii]$, элементы b_{κ} которого участвуют в аналитическом представлении сплайна следующим образом:

$$\sigma(t_1, \dots, t_n) = \sum_{\kappa=1}^N b_{\kappa} G_{m,n}(t_1 - a_{\kappa 1}, \dots, t_n - a_{\kappa n}) + \sum_{\kappa=1}^{ii} b_{N+\kappa} \prod_{l=1}^n t^{\alpha_{l\kappa}}.$$

Здесь

$$\begin{aligned}
 t_l &= \frac{x_l - MI}{MA}, \quad l = 1, \dots, n, \\
 b_\kappa &= B[\kappa], \quad \kappa = 1, \dots, N + ii, \\
 a_{\kappa j} &= A[\kappa, j], \quad \kappa = 1, \dots, N, \quad j = 1, \dots, n, \\
 \alpha_{l\kappa} &= J1[l, \kappa], \quad l = 1, \dots, n, \quad \kappa = 1, \dots, ii, \\
 G_{m,n}(t_{11} \dots t_n) &= \begin{cases} \left(\sum_{i=1}^n t_i^2 \right)^{m-n/2} \ln \left(\sum_{i=1}^n t_i^2 \right), & n - \text{четно}, \\ \left(\sum_{i=1}^n t_i \right)^{m-n/2}, & n - \text{нечетно}. \end{cases}
 \end{aligned}$$

Массив B вырабатывается обращением

$$SOL(H, R, N + ii, N, B);$$

Здесь H, N, ii имеют прежний смысл.

array $R[1 : N]$ — массив значений в узлах интерполяции.

array $B[1 : N + ii]$ — массив-результат, коэффициенты аналитического представления сплайна.

6.3.3. Процедура VAL

Процедура-функция VAL предназначена для расчета сплайна в заданной точке n -мерной области. Обращение имеет вид

$$Y := VAL(N, m, n, ii, C, B, J1, A, MA, MI);$$

Параметры N, m, n, ii имеют прежний смысл, массивы $J1, A$ и вещественные величины MA и MI выработаны процедурой $FORMH$, массив B выработан процедурой SOL .

array $C[1 : n]$ — массив координат точки, в которой рассчитывается сплайн-функция.

Перейдем теперь к описанию процедур $FORMH2, SOL2, VAL2$, которые осуществляют построение интерполяционной или сглаживающей сплайн-функции в n -мерной области при условии, что в хаотической системе узлов известны, быть может, не только значения функции, но и значения первых и вторых производных по заданным направлениям. Допускается, чтобы в одной точке задавалось несколько условий одного или различных типов (только значение функции, только значение

1-й производной, только значение 2-й производной, 1-я и 2-я, несколько 1-х по различным направлениям, несколько 2-х и т. д.). Непротиворечивость заданной информации о функции эффективно контролируется в процессе решения задачи.

6.3.4. Процедура *FORMH2*

Предназначена для формирования и обращения матрицы, возникающей в процессе решения задачи.

Обращение имеет вид

FORMH2(*N0*, *N1*, *N2*, *m*, *n*, *R0*, *R1*, *R2*, *ii*, *A0*, *A1*, *A2*, *A3*, *A4*, *A5*, *H*, *J1*, *MA*, *MI*, *ARM*);

integer *N0* > 0 — количество точек, в которых задано значение функции.

integer *N1* ≥ 0 — количество точек, в которых задано значение 1-й производной; если в какой-либо точке задано несколько значений 1-х производных по различным направлениям, точка участвует в списке несколько раз.

integer *N2* ≥ 0 — количество точек, в которых задано значение 2-й производной; если в точке задано несколько значений 2-х производных по различным направлениям, точку следует учитывать несколько раз.

integer *m* — параметр вариационного функционала.

integer *n* — размерность области.

real *R0*, *R1*, *R2* ≥ 0 — параметры сглаживания при значениях функции, 1-х производных и 2-х. Точнее, минимизации подвергается функционал

$$\Phi(u) = \int_{\Omega} \sum_{|\alpha|=m} (D^{\alpha}u)^2 d\Omega + R0 \sum_{P \in \pi_0} [u(P) - f_P]^2 + \\ + R1 \sum_{P \in \pi_1} \left[\frac{\partial u}{\partial \eta_P}(P) - f'_P \right]^2 + R2 \sum_{P \in \pi_2} \left[\frac{\partial^2 u}{\partial \zeta_P^2}(P) - f''_P \right]^2,$$

π_0 , π_1 , π_2 — множества точек, в которых задана функция, 1-я и 2-я производные; при увеличении параметров *R0*, *R1*, *R2* интерполяционным условиям приписывается все больший вес. Программа устроена так, что если положить *R0*, или *R1*, или *R2* равными 0, то в соответствующем члене будет осуществляться режим интерполяции, а в других — сглаживания. Если *R0* = *R1* = *R2* = 0, то работает чистый режим интерполяции.

integer ii — количество членов в полиноме степени $(m-1)$ от n переменных, $ii = (n+m-1)!/n!/(m-1)!$ подсчитывается пользователем.

array $A0[1:N0, 1:n]$ — массив координат точек, в которых известны значения функций.

array $A1[1:N1, 1:2 \times n]$ — массив, в котором хранятся координаты и направляющие косинусы, определяющие вектор, вдоль которого известно значение 1-й производной. Точнее, массив $A1[i, 1:2 \times n]$ в первых n позициях содержит координаты i -й точки, в следующих n позициях — направляющие косинусы, сумма квадратов которых равна 1. Вычисляется косинус угла между направлением и положительно направленной осью x_k .

array $A2[1:N2, 1:2 \times n]$ — аналогичный массив координат и направляющих косинусов для точек, в которых известны значения 2-х производных по направлению.

array $A3[1:N0, 1:n]$, $A4[1:N1, 1:2 \times n]$, $A5[1:N2, 1:n \times (n+3)/2]$ — массивы, вырабатываемые процедурой *FORMH2*.

array $H[1:N0+N1+N2+ii, 1:N0+N1+N2+ii]$ — массив-результат работы процедуры *FORMH2*.

integer array $J1[1:n, 1:ii]$ — массив, вырабатываемый *FORMH2*, смысл которого поясним при описании *SOL2*.

real MA, MI — величины, определяющие параметры линейной замены переменных, см. п. 6.3.1, переводящей все узловые точки в единичный куб.

Если задаваемая Вами информация о сплайне противоречива или количество условий недостаточно для его построения, на печать выдается текст 'ВЫРОЖДЕНА' и управление передается по метке *ARM*.

На параметры задачи должны выполняться следующие ограничения:

$$m > n/2 + \kappa, N0 + N1 > n, N0 + N1 + N2 \geq ii,$$

$$\kappa = \begin{cases} 0, & N1 = N2 = 0, \\ 1, & N1 \neq 0, N2 = 0, \\ 2, & N2 \neq 0. \end{cases}$$

6.3.5. Процедура *SOL2*

Процедура *SOL2* предназначена для вычисления коэффициентов аналитического представления сплайна. Именно процедура *SOL2* вырабатывает массив $B[1:N0+N1+N2+ii]$, элементы b_k которого участвуют в ана-

литическом представлении сплайна следующим образом:

$$\begin{aligned} \sigma(t_1, \dots, t_n) = & \sum_{\kappa=1}^{N_0} b_{\kappa} G_{m,n}(t_1 - a_{\kappa 1}^{(0)}, \dots, t_n - a_{\kappa n}^{(0)}) + \\ & + \sum_{\kappa=N_0+1}^{N_0+N_1} b_{\kappa} \frac{\partial G_{m,n}}{\partial \eta_{\kappa}}(t_1 - a_{\kappa 1}^{(1)}, \dots, t_n - a_{\kappa n}^{(1)}) + \\ & + \sum_{\kappa=N_0+N_1+1}^{N_0+N_1+N_2} b_{\kappa} \frac{\partial^2 G_{m,n}}{\partial \xi_{\kappa}^2}(t_1 - a_{\kappa 1}^{(2)}, \dots, t_n - a_{\kappa n}^{(2)}) + \\ & + \sum_{\kappa=1}^{ii} b_{N_0+N_1+N_2+\kappa} \prod_{l=1}^n t_i^{\alpha_{l\kappa}}. \end{aligned}$$

Здесь

$$t_i = \frac{x_i - MI}{MA}, \quad i = 1, \dots, n,$$

$$b_{\kappa} = B[\kappa], \quad \kappa = 1, 2, \dots, N_0 + N_1 + N_2 + ii,$$

$$a_{\kappa j}^{(0)} = A3[\kappa, j], \quad \kappa = 1, \dots, N_0, \quad j = 1, \dots, n,$$

$$a_{\kappa j}^{(1)} = A4[\kappa, j], \quad \kappa = 1, \dots, N_1, \quad j = 1, \dots, n,$$

$$a_{\kappa j}^{(2)} = A5[\kappa, j], \quad \kappa = 1, \dots, N_2, \quad j = 1, \dots, n,$$

$$\alpha_{l\kappa} = J1[l, \kappa], \quad l = 1, \dots, n, \quad \kappa = 1, \dots, ii,$$

$$\frac{\partial G_{m,n}}{\partial \eta_{\kappa}} = \sum_{i=1}^n A4[\kappa, n+i] \frac{\partial G_{m,n}}{\partial t_i}, \quad \kappa = 1, \dots, N_1,$$

$$\frac{\partial^2 G_{m,n}}{\partial \xi_{\kappa}^2} = \sum_{i=1}^n \sum_{j=i}^n A5[\kappa, n \times i + j - i(i-1)/2] \frac{\partial^2 G_{m,n}}{\partial t_i \partial t_j},$$

$$G_{m,n}(t_1, \dots, t_n) = \begin{cases} \left(\sum_{i=1}^n t_i^2 \right)^{m-n/2} \ln \left(\sum_{i=1}^n t_i^2 \right), & n - \text{четно}, \\ \left(\sum_{i=1}^n t_i^2 \right)^{m-n/2}, & n - \text{нечетно}. \end{cases}$$

Массив B вырабатывается обращением

$$SOL2(H, R, N_0, N_1, N_2, N_0 + N_1 + N_2 + ii, MA, B);$$

Массив H выработан $FORMH2$, N_0 , N_1 , N_2 , ii , MA имеют прежний смысл.

array $R[1 : N_0 + N_1 + N_2]$ — массив, содержащий в первых N_0 позициях значения функции, в следующих

$N1$ позициях значения 1-х производных, далее — 2-х производных.

`array B[1 :  $N0 + N1 + N2 + ii$ ]` — массив-результат работы *SOL2*.

6.3.6. Процедура-функция *VAL2*

Вещественная процедура *VAL2* предназначена для вычисления сплайн-функции в любой точке $x = (x_1, x_2, \dots, x_n)$. Обращение к ней имеет вид

$$Y := VAL2(N0, N1, N2, m, n, ii, C, B, J1, A3, A4, A5, MA, MI);$$

Параметры $N0, N1, N2, m, n, ii$ имеют прежний смысл, массивы $J1, A3, A4, A5$ и числа MA, MI выработаны процедурой *FORMH2*, массив B выработан процедурой *SOL2*.

`array C[1 :  $n$ ]` — координаты заданной пользователем точки, в которой производится расчет значения сплайна.

6.3.7. Рекомендуемый режим работы с комплектом *GREEN*

Как уже отмечалось, основные вычислительные затраты при решении задач с помощью комплекта *GREEN* приходится на обращение плотной матрицы. Требуется как минимум $(N0 + N1 + N2 + ii)^2$ ячеек оперативной памяти и $2/3 (N0 + N1 + N2 + ii)^3$ арифметических операций. Поэтому комплект *GREEN* наиболее рентабелен при решении серии задач на одной и той же хаотической сетке, но с разными входными данными. Матрицу H можно один раз вычислить, после чего многократно использовать, затрачивая на каждый вариант только $(N0 + N1 + N2 + ii)^2$ действий.

Далее, процедура *SOL* (или *SOL2*) вырабатывает коэффициенты аналитического представления сплайна. Однако формулы для вычисления такого представления все-таки достаточно сложны — содержат в себе логарифмы, корни, количество слагаемых увеличивается с ростом количества точек. Поэтому рекомендуемый режим работы — пересчет на удобную прямоугольную (параллелепипедальную) сетку, для которой существуют экономичные способы интерполяции, стандартные методы расчета полей изолиний и т. д.

6.3.8. Качественный тест комплекта *GREEN*

Продemonстрируем качество работы комплекта *GREEN* на примере. Зададим на единичном квадрате $[0, 1] \times [0, 1]$ аналитическую функцию

$$\varphi_0(x, y) = 3x^2 \sin(5y + x) + 4(y - x) \cos(3x + 2y).$$

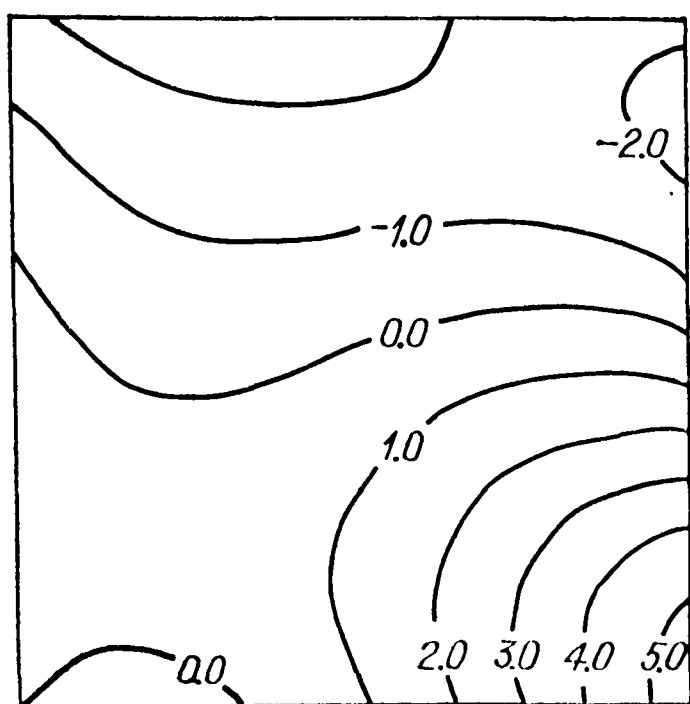


Рис. 10.

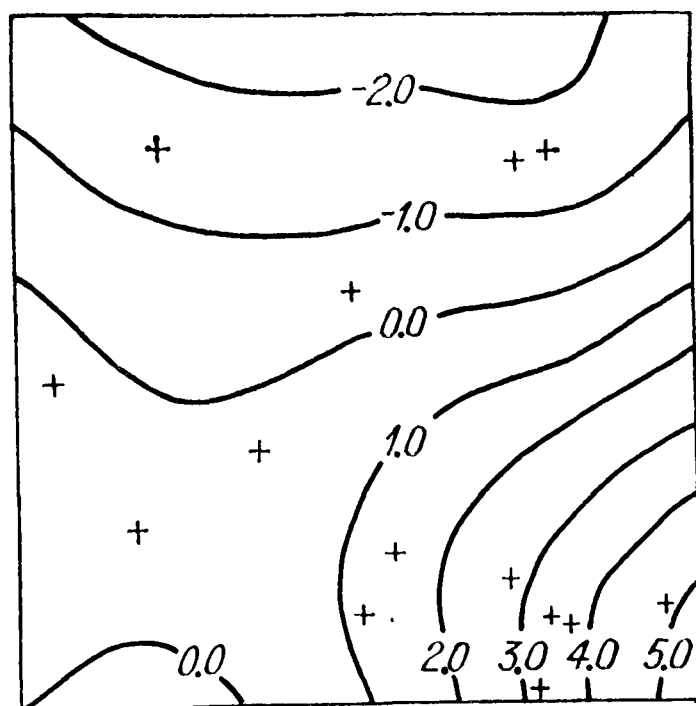


Рис. 11.

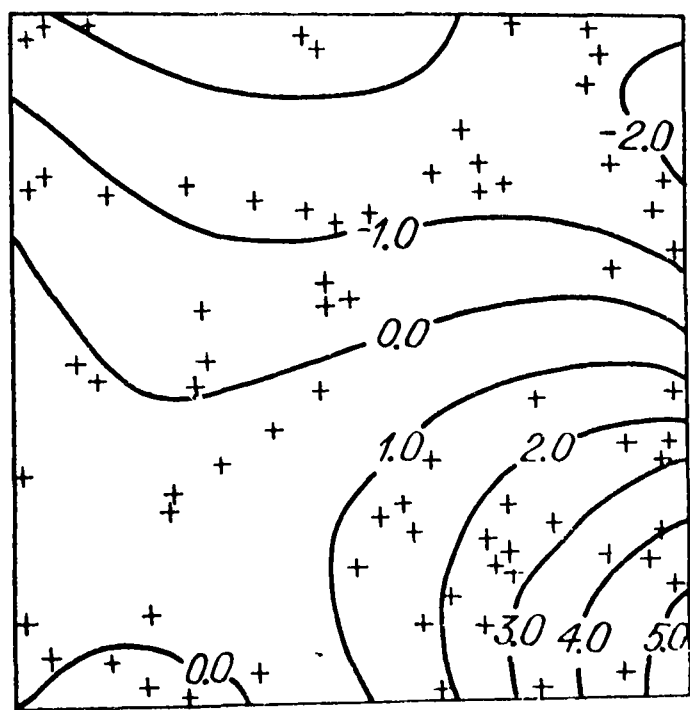


Рис. 12.

На рис. 10 показано поле изолиний этой функции. С помощью датчика случайных чисел нанесем на этот квадрат интерполяционные узлы. Их положение показано «крестиками» на рис. 11, 12. В первом случае их 20, во втором — 80. Вычислим значения функции в этих точках и произведем интерполяцию с помощью процедур *FORMH*, *SOL*, *VAL* при $m = 3$. Пересчитаем значения в узлы равномерной квадратной сетки 10×10 и вычертим поле

изолиний бикубической интерполяции такой сеточной функции. Как видно на рис. 11 (20 точек) и 12 (80 точек), при 80 узлах поле практически неотлично от рис. 10 и даже при 20 точках весьма мало отличается от исходного, рис. 10.

6.3.9. Отладочный тест процедур *FORMH*, *SOL*, *VAL*

Так как формулы для расчета сплайнов существенно отличаются для четных и нечетных n , произведем отдельно тестирование в двумерном ($n = 2$) и трехмерном ($n = 3$) случаях при $m = 2$.

Пусть $n = 2$.

На плоскости рассмотрим 5 интерполяционных узлов, координаты которых сведены в табл. 6.3.1, $A0[1:5, 1:2]$.

Ясно, что $ii = 3!/2!/1! = 3$. Будем проверять сразу режим интерполяции ($R0 = 0$) и режим сглаживания ($R0 = 5$). Обратимся дважды к процедуре *FORMH*. Будем выводить 1-ю строку матрицы H , массив $H[1,1:8]$. Результат см. в табл. 6.3.2.

Кроме массива H , результатом работы процедуры *FORMH* являются массив $J1[1:2, 1:3]$ (см. табл. 6.3.3), массив $A[1:5, 1:2]$ (см. табл. 6.3.4) и величины MA и MI ,

$$MA = 1.366\ 676\ 7, MI = -0.666\ 676\ 7.$$

Содержимое $J1$ и A не зависит от величины $R0$.

Зададим в узлах сетки значения $R[1:5]$, см. табл. 6.3.5.

Обратимся дважды к процедуре *SOL* с массивами H , полученными при $R0 = 0$ и $R0 = 5$. Получим массивы $B[1:8]$, см. табл. 6.3.6.

Далее, с помощью процедуры *VAL* вычислим значения интерполяционного ($R0 = 0$) и сглаживающего ($R0 = 5$) сплайнов в точках

$$P_1 = (-2/3, -1/3), P_2 = (-0.5, 0.5).$$

Результаты сведены в табл. 6.3.7.

Теперь решим трехмерную задачу, $n = 3$, $m = 2$. Ясно, что $ii = 4!/3!/1! = 4$. Рассмотрим 5 пространственных точек ($A0[1:5, 1:3]$), координаты которых сведены в табл. 6.3.8.

Опять будем одновременно проверять интерполяционный ($R0 = 0$) и сглаживающий ($R0 = 5$) режимы. Обратимся дважды к процедуре *FORMH* и выведем строки $H[1,1:9]$, см. табл. 6.3.9.

Кроме того, результатом работы *FORMH* являются числа

$$MA = 1.366\ 676\ 7, MI = -0.666\ 676\ 7,$$

массив $A[1:5,1:3]$, см. табл. 6.3.10, и целочисленный массив $J1[1:3,1:4]$, см. табл. 6.3.11.

В узлах сетки зададим значения функции $R[1:5]$ в соответствии с табл. 6.3.5. Обратимся дважды к процеду-

ре *SOL* с массивами *H*, полученными при $R0 = 0$ и $R0 = 5$. Массивы $B[1:9]$ расположены в табл. 6.3.12.

Вычислим с помощью *VAL* значения интерполяционного ($R0 = 0$) и сглаживающего ($R0 = 5$) сплайнов в точках

$$P_1 = (-2/3, -1/3, 1/3), P_2 = (-0.5, 0.5, 0.5).$$

Результаты сведены в табл. 6.3.13.

6.3.10. Отладочный тест процедур *FORMH2*, *SOL2*, *VAL2*

Положим $n = 2$, $m = 4$ ($m > n/2 + 2$). Тогда $ii = 5!/2!/3! = 10$. Пусть $N0 = N1 = N2 = 5$, $15 \geq ii$. Рассмотрим 5 точек, см. табл. 6.3.1. В каждой из них известно значение функции ($N0 = 5$), значения 1-й производной по направлению, определяемому направляющими косинусами $(1/\sqrt{2}, 1/\sqrt{2})$, и значения 2-х производных по направлению $(-1/\sqrt{2}, 1/\sqrt{2})$. Будем тестировать одновременно режим интерполяции ($R0 = R1 = R2 = 0$) и режим сглаживания ($R0 = 100$, $R1 = 10$, $R2 = 1$). Обратимся дважды к процедуре *FORMH2*, получим два массива вида $H[1:25, 1:25]$, при $i = 1$ его h_{ij} -элементы приведены в табл. 6.3.14 для некоторых j .

После окончания работы *FORMH2* $MA = 1.366\ 676\ 7$, $MI = -0.666\ 676\ 7$. Кроме этого, *FORMH2* вырабатывает массивы *A3*, *A4*, *A5*, *J1*. Для контроля приведем значения некоторых входящих в них элементов:

$$\begin{aligned} A3[2, 1] &= 0.6707, A4[2, 2] = 0.1220, \\ A4[2, 4] &= 0.7071, A5[2, 2] = 0.1220, \\ A5[2, 4] &= -1.0000, J1[1, 3] = 2.0000, J1[2, 5] = 1.0000. \end{aligned}$$

В узлах сетки зададим значения функции в соответствии с табл. 6.3.5. Значения 1-х и 2-х производных положим равными 0. Обратимся теперь дважды к процедуре *SOL2* и выработаем два массива вида $B[1:25]$. Значения некоторых элементов *B* приведены в табл. 6.3.15.

Теперь вычислим с помощью *VAL2* значения сплайнов в точках $P_1 = (-2/3, -1/3)$ и $P_2 = (-1/2, 1/2)$, см. табл. 6.3.16.

Решим теперь тестовую трехмерную ($n = 3$) задачу. По-прежнему $m = 4$ ($m > n/2 + 2$). Ясно, что $ii = 6!/3!/3! = 20$. Расположим в пространстве 8 точек: $P_1, P_2, \dots, P_8$, координаты которых заданы в табл. 6.3.17. В этой же таблице приведены значения функции f_i .

Пусть во всех 8 точках заданы 1-е производные по направлению $(1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})$ и 2-е производные по направлению $(-1/\sqrt{3}, 1/\sqrt{3}, -1/\sqrt{3})$. Значения 1-х и 2-х производных положим равными 0. Итак, $N0 = N1 = N2 = 8$. Будем тестировать только режим интерполяции (при условии, что Вы успешно протестировали режим сглаживания при $n = 2$). Обратимся к *FORMH2*. Результатом является массив $H[1:44, 1:44]$, некоторые h_{ij} -элементы которого приведены в табл. 6.3.18. Кроме того, результатом работы *FORMH2* являются $MA = 1.366\,676\,7$, $MI = -0.666\,676\,7$, целый массив $J1[1:3, 1:20]$, массивы $A3[1:8, 1:3]$, $A4[1:8, 1:6]$, $A5[1:8, 1:9]$. Приведем содержимое некоторых элементов этих массивов:

$J1[2, 8] = 1.0000$, $A3[5, 2] = 0.1951$, $A4[5, 2] = 0.1951$,
 $A4[5, 5] = 0.5774$, $A5[5, 2] = 0.1951$, $A5[5, 5] = -0.6667$.

Далее, обратимся к процедуре *SOL2* и выработаем массив $B[1:44]$. Значения некоторых его элементов приведены в табл. 6.3.19.

Наконец, с помощью процедуры *VAL2* вычислим значения сплайна в точках $P_1 = (-2/3, -1/3, 1/3)$ и $P_2 = (-1/2, 1/2, 1/2)$:

$$\begin{aligned}\sigma(P_1) &= -0.9999, \\ \sigma(P_2) &= 35.048.\end{aligned}$$

6.3.11. Процедура выбора параметра сглаживания *BESFOR*

Процедура *BESFOR* предназначена для выбора параметра сглаживания $R0 = 1/\alpha$ по критерию невязки, см. § 4.4 при построении сглаживающих сплайнов с помощью процедур *FORMH* и *SOL*. Если обозначить

$$\varphi(\alpha) = \sqrt{\frac{1}{N} \sum_{i=1}^N (\sigma_{\alpha}(x_i) - f_i)^2}$$

и ввести уровень невязки $\varepsilon > 0$, то параметр $R0$ считается оптимальным (см. п. 6.1.7) при

$$|\varphi(\alpha) - \varepsilon| \leq \varepsilon/10.$$

Обращение к процедуре имеет вид

BESFOR($N, m, n, R0, ii, A0, A, J1,$
 MA, MI, ARM, EPS, R, B);

Параметры N , m , n , ii , $A0$, A , $J1$, MA , MI совпадают с одноименными параметрами процедуры *FORMH*, а R и B — с параметрами процедуры *SOL*.

real $EPS > 0$ — заданный уровень невязки.

real $R0 > 0$ — перед началом работы процедуры *BESFOR* в $R0$ задается начальное приближение параметра сглаживания. После работы в $R0$ находится найденный оптимальный параметр.

label ARM — метка, на которую передается управление в случае, если EPS , заданное пользователем, слишком велико. Это значит, что уравнение $\varphi(\alpha) = \varepsilon$ не имеет решения, а решением задачи сглаживания является полином степени $(m - 1)$. Работа процедуры на этом завершается, коэффициенты при членах сглаживающего полинома засылаются в последние ii позиций массива B (см. процедуру *SOL*), а все остальные элементы в массиве B полагаются равными 0. При этом на печать выдается текст 'РЕШЕНИЕМ ЯВЛЯЕТСЯ ПОЛИНОМ' и величина максимального $\varepsilon_{кр} = EKP$, для которого существует решение $\varphi(\alpha) = \varepsilon$. Она печатается в виде " $EKP = \dots$ " и, кроме того, засылается в параметр EPS . Затем на печать выводится массив B . Управление передается на эту метку и в случае, описанном в процедуре *FORMH*.

Итак, после работы процедуры *BESFOR* мы получим сглаживающий сплайн или полином, являющийся решением задачи сглаживания. Для нахождения значения сплайна или полинома в произвольной точке можно использовать процедуру *VAL*. При этом величина $R0$ уже не важна. Однако в случае работы с серией однотипных задач можно, вычислив один раз $R0$, использовать его для решения всех остальных, применяя процедуры *FORMH* и *SOL*.

6.3.12. Тест процедуры *BESFOR*

Процедура *BESFOR* существенных отличий для четных и нечетных n не имеет, поэтому тестирование произведем в двумерном случае ($n = 2$) при $m = 2$.

Рассмотрим 5 интерполяционных узлов, координаты которых сведены в табл. 6.3.1, $A0[1:5, 1:2]$, со значениями в узлах сетки $R[1:5]$, см. табл. 6.3.5. Положим $EPS = 0.01$ и $R0 = 1$. Обращение

BESFOR(5, 2, 2, 1, 3, $A0$, A , $J1$, MA ,
 MI , ARM , 0.01, R , B);

приводит к оптимальному значению

$$R0 = 173.1845.$$

В тех же условиях при $EPS = 1$ решением будет полином, а $EKP = 0.93298$.

6.3.13. Процедура выбора параметров сглаживания *BESFO2*

Процедура *BESFO2* предназначена для выбора параметров сглаживания по критерию невязки из § 4.4 при построении сглаживающих сплайнов с помощью процедур *FORMH2* и *SOL2*. Процедура *BESFO2* позволяет производить сглаживание как самой функции, так и ее 1-х и 2-х производных и соответственно имеет три параметра сглаживания: $R0$, $R1$, $R2$, см. п. 6.3.4. Это вызвано тем, что, как правило, значения функции и ее 1-х и 2-х производных известны с различной степенью точности. Ввиду того, что необходимо найти три параметра сглаживания, в то время как способ нахождения теоретически обоснован лишь для одного, сделаем следующее упрощение. Используя априорную информацию о возможных отклонениях функции и производных, пользователь должен задать начальное приближение параметров сглаживания $R0_n$, $R1_n$ и $R2_n$. Тогда итоговые параметры сглаживания будут иметь вид $R0 = R0_n \times R$, $R1 = R1_n \times R$ и $R2 = R2_n \times R$, где $R = 1/\alpha$ находим по критерию невязки из § 4.4; другими словами, отношение $R0 : R1 : R2$ сохраняется. Обозначив за

$$\varphi(\alpha) = \left(\frac{1}{N0 \times R0_n} \sum_{i=1}^{N0} (\sigma_{\alpha}(x_i) - f_i)^2 + \frac{1}{N1 \times R1_n} \sum_{i=1}^{N1} (\sigma'_{\alpha, \eta_i}(y_i) - f'_i)^2 + \frac{1}{N2 \times R2_n} \sum_{i=1}^{N2} (\sigma''_{\alpha, \xi_i}(z_i) - f''_i)^2 \right)^{1/2}$$

и введя уровень невязки $\varepsilon > 0$, будем считать α оптимальным при

$$|\varphi(\alpha) - \varepsilon| \leq \varepsilon/10.$$

Заметим, что возможен случай, когда один или два параметра сглаживания равны 0, например, можно интерполировать функцию и сглаживать ее производные. При этом в формуле, определяющей $\varphi(\alpha)$, выпадут слагаемые с $R0 = 0$.

Обращение к процедуре имеет вид

$BESFO2(N0, N1, N2, m, n, R0, R1, R2, ii, A0, A1, A2, A3, A4, A5, J1, MA, MI, ARM, EPS, R, B);$

Параметры $N0, N1, N2, m, n, ii, A0, A1, A2, A3, A4, A5, J1, MA, MI$ совпадают с одноименными процедуры $FORMH2$, а R и B — соответственно процедуры $SOL2$.

$real R0, R1, R2$ — перед началом работы процедуры $BESFO2$ в них задаются начальные приближения параметров сглаживания функции и ее 1-х и 2-х производных соответственно. После работы процедуры в них находятся найденные оптимальные параметры. Напомним, что отношения пропорциональности у найденных параметров равны отношениям у начальных.

$real EPS$ — заданный уровень псевязки.

$label ARM$ — метка, на которую передается управление программой в случаях, описанных в процедурах $BESFOR$ и $FORMH2$, при этом на печать выводится та же информация.

После работы процедуры $BESFO2$ мы получим сглаживающий сплайн или полином. Для нахождения его значений в произвольной точке можно использовать процедуру $VAL2$. В случае работы с серией однотипных задач можно, вычислив один раз $R0, R1$ и $R2$, использовать их для решения всех остальных посредством процедур $FORMH2$ и $SOL2$.

6.3.14. Тест процедуры $BESFO2$

Процедура $BESFO2$ также не имеет существенных отличий при четных и нечетных n , поэтому тестирование произведем в двумерном случае, точно таком же, как и при тестировании процедуры $FORMH2$, см. п. 6.3.10. Зададим $EPS = 0.1$ и начальные $R0 = 1, R1 = 5, R2 = 25$.

Обращение

$BESFO2(5, 5, 5, 4, 2, 1, 5, 25, 10, A0, A1, A2, A3, A4, A5, J1, MA, MI, ARM, 0.1, R, B);$

приводит к оптимальным значениям: $R0 = 246.7, R1 = 1233.6, R2 = 6167.8$. В тех же условиях при $EPS = 1$ решением будет полином, а $EKP = 0.8528$.

§ 6.4. КОМПЛЕКТ *AVERAGE* ДЛЯ ФИЛЬТРАЦИИ ЗАШУМЛЕННЫХ ДАННЫХ

Комплект *AVERAGE* состоит из двух основных процедур и одной вспомогательной. Он предназначен для построения кусочно-постоянных осредняющих функций, сохраняющих (или уничтожающих) при осреднении заданные пользователем системы квазимногочленов вида $\varepsilon_\kappa(t) = t^{n_\kappa} e^{(\eta_\kappa + i\zeta_\kappa)t}$ и, как всякое осреднение, подавляющих высокочастотные помехи на экспериментальной кривой (см. гл. 5).

6.4.1. Описание комплекта *AVERAGE*

Пусть на оси t (рис. 13) с равномерным шагом $h > 0$ процедура-функция $F(j)$ (j — номер узла сетки) генерирует сеточную функцию. Предположим, что мы хотим сохранить (или уничтожить) при осреднении квазиполиномы

$$\varepsilon_\kappa(t) = t^{n_\kappa} (\cos \zeta_\kappa t + i \sin \zeta_\kappa t) e^{\eta_\kappa t}, \quad \kappa = 1, 2, \dots, n.$$

Пусть зона постоянства осредняющей функции содержит w шагов сетки. Величину $w \geq 1$ задает пользователь. Изменим величины $\zeta_\kappa, \eta_\kappa$ по формулам гл. 5,

$$\begin{aligned} \tilde{\eta}_\kappa &= h \cdot w \cdot \eta_\kappa, \\ \tilde{\zeta}_\kappa &= h \cdot w \cdot \zeta_\kappa, \quad \kappa = \overline{1, n}. \end{aligned}$$

Сформируем массив $Z[1:4, 1:n]$ следующим образом:

$$\begin{aligned} Z[1, \kappa] &= n_\kappa, \\ Z[2, \kappa] &= \tilde{\eta}_\kappa, \\ Z[3, \kappa] &= \tilde{\zeta}_\kappa, \end{aligned}$$

$$Z[4, \kappa] = \begin{cases} 1, & \text{если квазиполином следует сохранить,} \\ 0, & \text{если его следует уничтожить,} \end{cases}$$

$\kappa = 1, 2, \dots, n$. Подсчитаем дифференциальную размерность системы квазиполиномов (см. гл. 5) по формуле

$$p = \sum_{\kappa=1}^n \omega_\kappa, \quad \omega_\kappa = \begin{cases} n_\kappa + 1, & \zeta_\kappa = 0, \\ 2(n_\kappa + 1), & \zeta_\kappa \neq 0. \end{cases}$$

Теперь рассчитаем коэффициенты осредняющей функции $\alpha_1, \alpha_2, \dots, \alpha_p$. Это делается с помощью процедуры *COMBI*, обращение к которой имеет вид

$$\text{COMBI}(Z, N, AL, P);$$

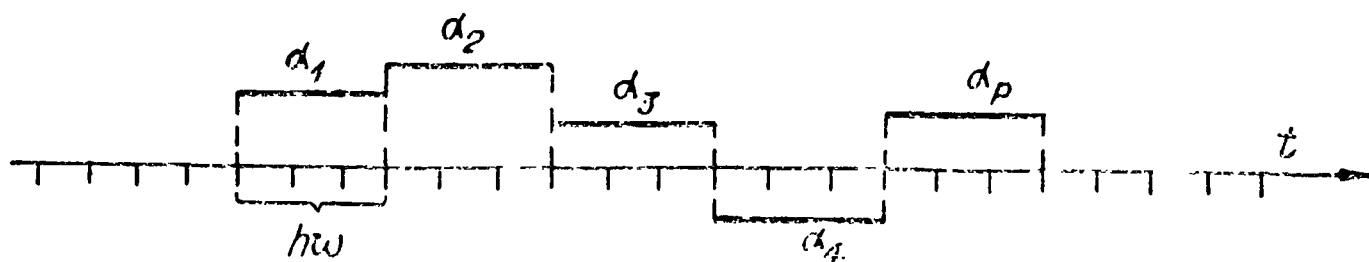


Рис. 13.

integer N — количество D -независимых квазиполиномов.

array $Z[1:4, 1:N]$ — массив, который мы описали выше.

array $AL[1:P]$ — массив коэффициентов осредняющей функции, результат работы.

integer P — D -размерность исходной системы квазиполиномов.

В процессе работы *COMBI* осуществляется контроль некоторых параметров. Если пользователь неверно задает P , D -размерность системы, на печать выдается текст 'ПАРАМЕТР P ЗАДАН НЕВЕРНО. ЗАДАЙТЕ $P = \dots$ ' и работа процедуры прекращается.

Если заданная пользователем система квазимногочленов D -зависима, то на печать выдается текст 'СИСТЕМА КВАЗИМОНОМОВ D -ЗАВИСИМА' и работа процедуры прекращается.

Для решения системы алгебраических уравнений в процедуре *COMBI* используется процедура *MVR* метода вращений, ранее описанная в п. 6.1.5.

Вторая процедура комплекта *AVER* предназначена для построения осреднения, если коэффициенты осредняющей функции уже известны. Обращение к ней имеет вид

$$AVER(F, N0, NT, P, w, AL, FCL);$$

real procedure F — имя процедуры-функции целого аргумента, обращение к которой имеет вид $x := F(j)$. Эта процедура генерирует значения сеточной функции в узле с номером j . Для правильной работы *AVER* необходимо, чтобы такие значения могли быть вычислены при j из интервала от $N0 - P \times w/2 - 2$ до $NT + P \times w/2 + 2$.

integer $N0, NT$ — начальный и конечный номера узлов сетки, в которых мы хотим вычислить значения осредненной функции.

integer $P \geq 1$ — количество осредняющих коэффициентов, выработанных ранее процедурой *COMBI*.

integer $w \geq 1$ — величина, регулирующая количество узлов сетки, укладываемых в зону постоянства осред-

няющей функции. Величиной $h \times w \times P$ определяется ширина носителя осредняющей функции.

array $AL[1:P]$ — массив коэффициентов осредняющей функции, выработанный процедурой *COMBI*.

array $FCL[N0:NT]$ — результаты работы процедуры *AVER*, значения осредненной сеточной функции в узлах сетки, начиная с $N0$ до NT .

Наконец, комплект *AVERAGE* содержит вспомогательную процедуру *INDIC*, которая в случае $n_1 = n_2 = \dots = n_n = 0$ позволяет вычислять в любой точке комплексной плоскости $z = a + bi$ значение комплексной функции $A_{p,\alpha}(z)$, см. § 5.7, т. е.

$$K_p \widehat{*} e^{(a+bi)t} = A_{p,\alpha}(a + bi) \times e^{(a+bi)t}.$$

По поведению функции $A_{p,\alpha}(z)$ можно судить, что произошло при осреднении функции e^{zt} с помощью построенного Вами интегрального оператора (что случилось с гармониками промежуточных частот, экспонентами и т. д.).

Обращение к процедуре имеет вид

$$INDIC(P, AL, a, b, Re, Im);$$

Параметры P, AL имеют прежний смысл.

real a, b — соответственно вещественная и мнимая части точки z , в которой подсчитывается значение $A_{p,\alpha}(z)$.

real Re, Im — вещественная и мнимая части функции $A_{p,\alpha}(z)$.

6.4.2. Пример работы с комплектом *AVERAGE* и отладочный тест

Пусть на отрезке $[-2, 4]$ с шагом $h = 0.02$ задана экспериментальная сеточная функция. Поместим ее значения в массив $A[-100:200]$. Организуем процедуру-функцию $F(j)$, которая по номеру j выдает значение $A[j]$, $-100 \leq j \leq 200$.

Пусть известно, что кривая содержит в качестве информативной части произвольную линейную комбинацию функций вида $1, \sin 5t, t^2 e^{1.5t}$ (или их производных) и заданную функцию «помехи» вида $\sin 10t$, которая должна быть уничтожена (например, это фон питающей сети в радиотехническом устройстве). Кроме того, кривая содержит другие экспериментальные помехи, которые убывают в процессе осреднения.

Итак, в нашем случае:

$$\begin{aligned} n_1 &= 0, & \eta_1 &= 0, & \xi_1 &= 0 & (\text{сохранить}), \\ n_2 &= 0, & \eta_2 &= 0, & \xi_2 &= 5 & (\text{сохранить}), \\ n_3 &= 2, & \eta_3 &= 1.5, & \xi_3 &= 0 & (\text{сохранить}), \\ n_4 &= 0, & \eta_4 &= 0, & \xi_4 &= 10 & (\text{уничтожить}). \end{aligned}$$

В соответствии с правилами, $\tilde{\eta}_k = h \cdot w \cdot \eta_k$, $\tilde{\xi}_k = h \cdot w \cdot \xi_k$.
Выберем $w = 10$; тогда $h \cdot w = 0.02 \times 10 = 0.2$ и

$$\begin{aligned} n_1 &= 0, & \tilde{\eta}_1 &= 0, & \tilde{\xi}_1 &= 0, \\ n_2 &= 0, & \tilde{\eta}_2 &= 0, & \tilde{\xi}_2 &= 1.0, \\ n_3 &= 2, & \tilde{\eta}_3 &= 0.3, & \tilde{\xi}_3 &= 0, \\ n_4 &= 0, & \tilde{\eta}_4 &= 0, & \tilde{\xi}_4 &= 2. \end{aligned}$$

Заполним этими величинами массив $Z[1:4, 1:4]$, причем $Z[4, 1] = Z[4, 2] = Z[4, 3] = 1$, $Z[4, 4] = 0$. Далее, вычислим D -размерность системы квазимногочленов

$$P = 1 + 2 + 3 + 2 = 8.$$

Обратившись к процедуре *COMBI*, найдем массив $AL[1:8]$:

$$\begin{aligned} AL[1] &= 0.103\,327\,852\,6, \\ AL[2] &= -0.351\,423\,039\,8, \\ AL[3] &= 0.410\,355\,645\,1, \\ AL[4] &= 0.234\,149\,497\,6, \\ AL[5] &= 0.642\,579\,919\,2, \\ AL[6] &= 0.036\,631\,456\,6, \\ AL[7] &= -0.101\,251\,197\,2, \\ AL[8] &= 0.025\,629\,865\,9. \end{aligned}$$

Уточним теперь кривую, которую предстоит осреднять. Рассмотрим функцию

$$f(t) = -0.5 + 2 \cos(5t + 3) + 0.05(t - 1)^2 e^{1.5t} - 0.1 \cos 10t$$

на отрезке $[-2, 4]$ с шагом $h = 0.02$. Произведем осреднение такой сеточной кривой от $N0 = 0$ до $NT = 100$, т. е. вычислим массив $FCL[0:100]$ с помощью обращения

$$AVER(F, 0, 100, 8, 10, AL, FCL).$$

При этом

$$\begin{aligned} FCL[0] &= -2.428\,338, & FCL[1] &= -2.447\,125, \\ FCL[50] &= -0.790\,742. \end{aligned}$$

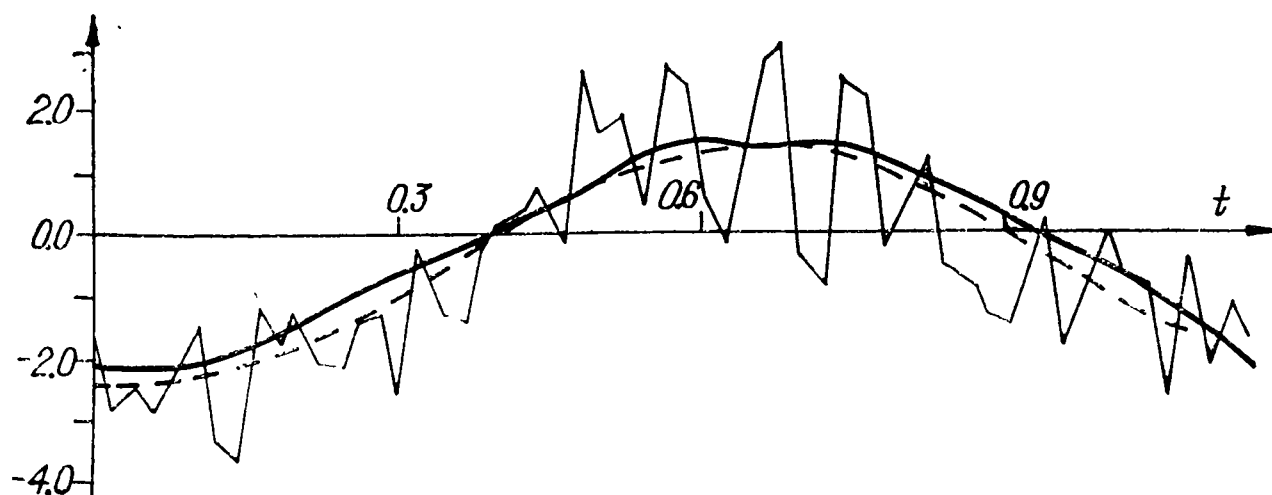


Рис. 14.

Теперь произведем осреднение кривой в «реальной обстановке», когда она зашумлена экспериментальной помехой. С помощью датчика случайных чисел добавим к каждому сеточному значению случайную величину, равномерно распределенную в интервале $[-2, 2]$. На рис. 14 изображены аналитическая кривая (пунктир)

$$g(t) = -0.5 + 2 \cos(5t + 3) + 0.05(t - 1)^2 e^{1.5t}$$

и зашумленная кривая (тонкая линия)

$$\tilde{f}(t) = f(t) + \zeta(t).$$

Как видите, ошибки за счет $\zeta(t)$ и $0.1 \cos 10t$ весьма значительны. На этом же рисунке изображен результат обработки. К сожалению, датчики случайных чисел на различных ЭВМ работают по-разному, поэтому на рис. 14 мы приводим только качественную картину.

ТЕКСТ КОМПЛЕКТА *ODD*

```

_procedure assh(n1,n2,x,p,h);
_begin
_integer i,j,k,n,l;
_array x1,f1(0:p1);
_if p>n2-n1_then _goto печать1;
n:=n2-n1+1;
_for k:=1_step 1_until n-p_do _begin
_for i:=0_step 1_until p_do _begin x1[i]:=x(n1+k-1+i);
f1[i]:=0_end ;
_for i:=0_step 1_until p_do _begin f1[i]:=1;
_for j:=1_step 1_until p_do
_for l:=0_step 1_until p-j-do
f1[l]:=(f1[l+1]-f1[l])/(x1[l+j]-x1[l]);
h[k,i+1]:=f1[0];
_for j:=0_step 1_until p-do f1[j]:=0;
_end ;
_end ;_goto лаб;
печать1:output('t','количество узлов сетки
недостаточно для построения интерполанта ','/');
лаб:_end assh;
_procedure assa(n1,n2,x,p,h,a1,a);
_begin
_integer i1,i2;
_integer i,j,n,k,l,m;_real s,c;
_integer fs;
fs:=(fac(p-1))↑2;
n:=n2-n1+1;
_begin _array xx(1:n);
_real _procedure kb(m,i,j);
_begin _integer k;

```

```

_real a,b,h,c,d;
h:=xx[m+1]-xx[m];
a:=((xx[i]-xx[m])+(xx[i]-xx[m+1]))/h;
b:=((xx[j]-xx[m])+(xx[j]-xx[m+1]))/h;
d:=0;
_for k:=1_step 1_until p_do _begin
c:=nw[p,k,1];
d:=d+((a-c)*(b-c))^(p-1)*nw[p,k,2]_end ;
kb:=d*(h/2)^(2*p-1);
_end kb;
_for k:=1_step 1_until n_do
xx[k]:=x[n1+k-1];
_for i:=1_step 1_until n-p_do
_for k:=0_step 1_until min(p-1,n-p-i)_do
_begin s:=0;

_for j:=i+k_step 1_until i+p-1_do
_for l:=j-i+2_step 1_until p+1_do
_for m:=j-i-k+2_step 1_until p+1_do
s:=s+h[i,l]*h[i+k,m]*
kb(j,i+l-1,i+k+m-1);
a[i+k,p-1-k]:=s/fs_end ;
_if abs(a1)>=10^-17_then
_begin
_for i:=1_step 1_until n-p_do
_for j:=p-1_step -1_until 0_do a[i,j+1]:=a[i,j];
_for i:=1_step 1_until n-p_do a[i,0]:=0;
_for i:=1_step 1_until n-p_do
_for j:=max(p+1-i,0)_step 1_until p_do
_begin s:=0;
_for k:=i_step 1_until i+j_do
s:=s+h[i,k-i+1]*h[i+j-p,k-i-j+p+1];

```

```

a[i,j]:=a[i,j]+a1xs_end ;
_end ;
_end ;
_end assa;

_procedure decomp(n1,n2,p,a1,a,l);
_begin
_real d1,d2;
_if abs(a1)≤10-17_then
chobanddet(n2-n1-p+1,p-1,a,l,d1,d2,печат2)
_else
chobanddet(n2-n1-p+1,p,a,l,d1,d2,печат2);
_goto lab;
печат2:output('t','ошибка во входных
данных процедуры decomp ','/');
lab:_end decomp;

_procedure solve(n1,n2,p,h,a1,r,l,f,φ);
_begin
_integer i,j,κ,n;_real s,c;
n:=n2-n1+1;
_begin _array b(1:n-p,1:r);
_for κ:=1_step 1_until r_do
_for i:=1_step 1_until n-p_do _begin
s:=0;_for j:=1_step 1_until p+1_do
s:=s+h[i,j]×f[i+j+n1-2,κ];
b[i,κ]:=s_end ;
_if abs(a1)≤10-17_then
chobandsol(n-p,p-1,r,l,b,φ)
_else _begin
chobandsol(n-p,p,r,l,b,φ);
_for κ:=1_step 1_until r_do
_for i:=1_step 1_until n-do _begin

```



```

s:=0;_for j:=p+1_step -1_until 1_do
_begin _if 1≤i-j+1^i-j+1≤n-p_then c:=h(i-j+1,j)x
φ(i-j+1,κ)
_else c:=0;s:=s+c_end ;
f(n1+l-1,κ):=f(n1+l-1,κ)-a1xs_end ;
_end ;_end ;
_end solve;

_procedure repres(n1,n2,p,x,f,r,φ,h,aa);
_begin
_integer i,j,n,κ,l,m,t,q;
_real s,c;
_integer fq,_real e;
_integer fp;
fp:=fac(p);
n:=n2-n1+1;
q:=2xp-1;
fq:=fac(q);
_begin _array xx[1:n],a1[1:p-1,1:p-1],b1[1:r,1:p-1];
_for i:=1_step 1_until n_do xx[i]:=x(n1+i-1);
_for i:=n1_step 1_until n2_do
_for j:=1_step 1_until r_do a1(i,0,j):=f(i,j);
_for m:=1_step 1_until r_do _begin
_for κ:=0_step 1_until p-1_do _begin
_for j:=1_step 1_until n-1_do _begin
s:=0;_for l:=max(1,j-p+1)_step 1_until min(n-p,j)_do
_for l:=1_step 1_until p+1_do _if j<i+l-1_then
s:=s+h(i,l)xφ(i,m)x(xx[i+l-1]-xx[j])^(p-1-κ);
a1(n1+j-1,p+κ,m):=s/fac(p-1-κ)x(-1)^κ;
_end ;a1(n1-1,p+κ,m):=a1(n2,p+κ,m):=0_end ;
_end m;
_for κ:=1_step 1_until n-1_do _begin

```

```

_if k+p-1≤n_then _begin
_for j:=k+1_step 1_until k+p-1_do
_for i:=1_step 1_until p-1_do
a1[j-k,i]:=(xx[j]-xx[k])i/fac(i)
_end _else _begin
_for j:=k-1_step -1_until k-p+1_do
_for i:=1_step 1_until p-1_do
a1[k-j,i]:=(xx[j]-xx[k])i/fac(i)
_end a1;
_for m:=1_step 1_until r_do _begin
_if k+p-1≤n_then _begin
_for j:=k+1_step 1_until k+p-1_do _begin
s:=c:=0;e:=xx[j]-xx[k];
_for i:=q_step -1_until p_do
s:=s×e/(i+1)+aa[n1+k-1,i,m];
s:=s×ep/fp;
_for i:=k+1_step 1_until j_do
c:=c+(aa[n1+i-1,q,m]-aa[n1+i-2,q,m])×
(xx[j]-xx[i])q;
b1[m,j-k]:=f[n1+j-1,m]-f[n1+k-1,m]-s-c/fq
_end j;_end _else _begin
_for j:=k-1_step -1_until k-p+1_do _begin
s:=c:=0;e:=xx[j]-xx[k];
_for i:=q_step -1_until p_do
s:=s×e/(i+1)+aa[n1+k-1,i,m];
s:=s×ep/fp;
_for i:=k-1_step -1_until j_do
c:=c+(aa[n1+i,q,m]-aa[n1+i-1,q,m])×
(xx[j]-xx[i+1])q;
b1[m,k-j]:=f[n1+j-1,m]-f[n1+k-1,m]-s+c/fq
_end j;_end b1;_end m;

```

```

mvr(a1,b1,p-1,r);
_for j:=1_step 1_until p-1_do
_for m:=1_step 1_until r_do
aa[n1+k-1,j,m]:=b1[m,j];
_end j;
c:=xx[n]-xx[n-1];
_for m:=1_step 1_until r_do _begin
_for i:=1_step 1_until p-1_do _begin
aa[n1-1,i,m]:=aa[n1,i,m];
s:=0;_for j:=q_step -1_until i_do
s:=sxc/(j-i+1)+aa[n2-1,j,m];
aa[n2,i,m]:=s_end ;
aa[n1-1,0,m]:=f[n1,m];
aa[n2,0,m]:=f[n2,m]_end ;
_end ;
_end repres;

_real _procedure fn(n1,n2,p,x,aa;k,x0);
_begin
_integer i,j;-real s,c;
_if x0<x[n1]_then _begin i:=n1-1;-goto lab2_end
_if x0>x[n2]_then _begin i:=n2;-goto lab2_end
i:=n1;
lab1:_if x[i]≤x0^x0≤x[i+1]_then
_goto lab2_else
_begin i:=i+1;-goto lab1_end ;
lab2:
s:=0;c:=x0-x[max(i,n1)];
_for j:=2×p-1_step -1_until 0_do
s:=sxc/(j+1)+aa[i,j,k];
fn:=s;
_end fn;

```

```

_procedure bespar(p,n1,n2,x,f,eps,al,lab);
_begin
_integer i,j,k,n;
n:=n2-n1+1;_begin
_real ek,s,t,m,q,phi1,phi2;
_array aa(1:p,1:p),bb(1:1,1:p),h(1:n-p,1:p+1),
l,a(1:n-p,0:p),phi,phi2(1:n-p,1:1),f1,f2(n1:n2,1:1);
aa(1,1):=n;t:=(x(n1)+x(n2))/2;m:=x(n2)-x(n1);
_for i:=0_step 1_until p-1_do
_for j:=(if i=0_then 1_else i)
_step 1_until p-1_do _begin
s:=0;_for k:=n1_step 1_until n2_do
s:=s+((x(k)-t)/m)^(i+j);
aa(j+1,i+1):=aa(i+1,j+1):=s_end ;
_for k:=n1_step 1_until n2_do
bb(1,1):=bb(1,1)+f(k,1);
_for i:=1_step 1_until p-1_do _begin
s:=0;_for k:=n1_step 1_until n2_do
s:=s+((x(k)-t)/m)^i*f(k,1);
bb(1,i+1):=s_end ;
mvr(aa,bb,p,1);
ek:=0;
_for k:=n1_step 1_until n2_do _begin
s:=(x(k)-t)/m;
q:=bb(1,p);
_for i:=p_step -1_until 2_do
q:=q*s+bb(1,i-1);
ek:=ek+(q-f(k,1))^2;
_end ;
ek:=sqrt(ek/(n2-n1+1));
_if eps>ek_then _goto lab;

```

```

me:
_for i:=n1_step 1_until n2_do
f1(i,1):=f(i,1);
assh(n1,n2,x,p,h);
assa(n1,n2,x,p,h,a1,a);
decom6(n1,n2,p,a1,a,l);
solve(n1,n2,p,h,a1,1,l,f1,φ);
φa1:=φ1a1:=0;
_for i:=n1_step 1_until n2_do
φa1:=(f(i,1)-f1(i,1))2+φa1;
φa1:=sqrt(φa1/n);
_if abs(φa1-eps)≤eps/10_then _begin
_for i:=n1_step 1_until n2_do
f1(i,1):=f1(i,1);_goto fin_end ;
_for i:=n1_step 1_until n2_do
f2(i,1):=(f1(i,1)-f(i,1))/a1;
solve(n1,n2,p,h,a1,1,l,f2,φ2);
_for i:=n1_step 1_until n2_do
f2(i,1):=f2(i,1)-(f1(i,1)-f(i,1))/a1;
_for i:=n1_step 1_until n2_do
φ1a1:=φ1a1+f2(i,1)×(f(i,1)-f1(i,1));
φ1a1:=φa1/a1-φ1a1/n/φa1;
a1:=a12×φ1a1/(-a1×φ1a1-φa1+φa12/eps);
_goto me;
fin:_end ;
_end bespar;

```

ТЕКСТ КОМПЛЕКТА *EVEN*

```

_procedure assh2(n1,n2,x,p,h);
_begin
_integer i,j,k,n,l,p1;
_array x1,f1[0:p+1];
_if p>n2-n1-1_then _goto печать1;
n:=n2-n1+1;p1:=p+1;
_for k:=1_step 1_until n-p1_do _begin
_for i:=0_step 1_until p1_do _begin
f1[i]:=0;x1[i]:=x[n1+k-1+i]_end ;
_for i:=0_step 1_until p_do _begin f1[i]:=1;
_for j:=1_step 1_until p1_do
_for l:=0_step 1_until p1-j_do
f1[l]:=(f1[l+1]-f1[l])/(x1[l+j]-x1[l]);
_if l≠0_then h[k,i+1]:=f1[0]+h[k,i]
_else h[k,1]:=f1[0];
_for j:=0_step 1_until p1_do f1[j]:=0;
_end ;
_for j:=1_step 1_until p1_do
h[k,j]:=-h[k,j]×(x1[j]-x1[j-1]);
_end ;_goto lab;
печать1:output('t','количество узлов сетки
недостаточно для построения
интерполанта такой степени','3/');
lab:_end assh2;

_procedure assa2(n1,n2,x,p,h,a1,a);
_begin
_real _procedure kb(m,i,j);
_begin
_integer k;_real a,b,h,c,d;
h:=xx[m+1]-xx[m];

```

```

a:=(2*xx[i]-xx[m+1]-xx[m])/h;
b:=(2*xx[j]-xx[m+1]-xx[m])/h;
d:=0;
_for k:=1_step 1_until p1_do _begin
c:=nwlp1,k,1);
d:=d+((a-c)*(b-c))p*nwlp1,k,2);_end ;
κb:=d*(h/2)(2*p+1);
_end κb;
_integer i1,i2,i,j,n,κ,l,m,p1;
_real s1,s2;
_real s;
_array xx(1:n2-n1+1);
_integer fs;fs:=(fac(p))2;
n:=n2-n1+1;p1:=p+1;
_for i:=1_step 1_until n_do xx(i):=x(n1+i-1);
_for i:=1_step 1_until n-p1_do
_for κ:=0_step 1_until min(p,n-p1-i)_do _begin
s:=0;
_for j:=i+κ_step 1_until i+p_do

_for l:=j+1_step 1_until i+p1_do _begin
_if l=i+p1_then s1:=h(i,p1)/(xx[l]-xx[l-1])_else
s1:=h(i,l-i)/(xx[l]-xx[l-1])-h(i,l-i+1)/(xx[l+1]-xx[l]);
_for m:=j+1_step 1_until i+κ+p1_do _begin
_if m=i+κ+p1_then s2:=h(i+κ,p1)/(xx[m]-xx[m-1])_else
s2:=h(i+κ,m-i-κ)/(xx[m]-xx[m-1])
-h(i+κ,m-i-κ+1)/(xx[m+1]-xx[m]);
s:=s+s1*s2*κb(j,l,m);
_end m;_end l;
a(i+κ,p-κ):=s/fs;
_end κ;
_if abs(a)≥10-17_then

```

```

_begin
_for i:=1_step 1_until n-p1_do
_for j:=max(p1-i,0)_step 1_until p_do _begin
s:=0;
_for k:=i_step 1_until i+j_do
s:=s+h(i,k-i+1)*h(i+j-p,k-i-j+p+1);
a(i,j):=a(i,j)+a1*s;_end ;_end ;
_end assa2;

_procedure decomp2(n1,n2,p,a,l);
_begin
_real d1,d2;
chobanddet(n2-n1-p,p,a,l,d1,d2,печат2);
_goto lab;
печат2:output('t','ошибка во входных данных
процедуры decomp2','/');
lab:_end decomp2;

_procedure solve2(n1,n2,p,n,a1,r,l,f,φ);
_begin
_integer i,j,k,n,p1;_real s,c;
n:=n2-n1+1;p1:=p+1;
_begin _array b(1:n-p1,1:r);
_for k:=1_step 1_until r_do
_for i:=1_step 1_until n-p1_do _begin
s:=0;_for j:=1_step 1_until p1_do
s:=s+h(i,j)*f(n1+i+j-2,k);
b(i,k):=s_end ;
chobandsol(n-p1,p,r,l,b,φ);
_if abs(a1)≥10-17_then
_begin
_for k:=1_step 1_until r_do
_for i:=1_step 1_until n-1_do _begin

```



```

s:=0;_for j:=p1_step -1_until 1_do
_begin _if 1≤i-j+1∧i-j+1≤n-p1_then
c:=h(i-j+1,j)×φ(i-j+1,κ)
_else c:=0;s:=s+c;_end ;
f(n1+i-1,κ):=f(n1+i-1,κ)-a1xs;_end ;_end ;_end ;
_end solve2;

_procedure repres2(n1,n2,p,x,f,r,φ,h,aa);
_begin
_array xx(1:n2-n1+1);
_integer i,j,κ,m,p1,n;
_integer l,q,fq,fp1;
_real s,ss,s1,s2;
_integer q1,c1;
q1:=2×p;
p1:=p+1;n:=n2-n1+1;
q:=2×p+1;fq:=fac(2×p+1);fp1:=fac(p1);
_for i:=1_step 1_until n_do xx(i):=x(n1+i-1);
_for κ:=1_step 1_until r_do
_for j:=0_step 1_until p_do _begin
_for m:=1_step 1_until n-1_do _begin s:=0;
_for i:=max(1,m-p1+1)_step 1_until min(n-p1,m)_do
_for l:=1_step 1_until p1_do
_begin
_if m<i+l-1_then
s:=s+h(i,l)×φ(i,κ)×((xx(i+l)-xx(m))f(p-j)-
(xx(i+l-1)-xx(m))f(p-j))/(xx(i+l)-xx(i+l-1));
_if m=i+l-1_then
s:=s+h(i,l)×φ(i,κ)×(xx(m+1)-xx(m))f(p-j-1);_end ;
aa(n1+m-1,j+p,κ):=s/fac(p-j)×(-1)fj;
_end m;
aa(n1-1,p+j,κ):=aa(n2,p+j,κ):=0;_end j;

```

```

_begin
_array c(1:p,1:p),b(1:r,1:p);
_real _procedure qu(k,l,j);
_begin
qu:=((xx(k+1)-xx(i))(j+1)-
(xx(k)-xx(i))(j+1))/(xx(k+1)-xx(k))/
fac(j+1)_end qu;
_for m:=1_step 1_until n-1_do _begin
_if m+p≤n_then _begin
_for i:=0_step 1_until p-1_do
_for j:=0_step 1_until p-1_do
c(i+1,j+1):=qu(m+i,m,j);
_for k:=1_step 1_until r_do
_for i:=0_step 1_until p-1_do _begin
s:=ss:=0;
_for l:=p_step 1_until q1_do
s:=s+aa(n1+m-1,l,k)×qu(m+i,m,l);
_for l:=m_step 1_until m+i+1_do
ss:=ss+(aa(n1+l,q1,k)-aa(n1+l-1,q1,k))×
qu(m+i,m+l,q1);
b(k,i+1):=f(n1+m+i-1,k)-s-ss;
_end _end _else _begin
_for i:=0_step 1_until p-1_do
_for j:=0_step 1_until p-1_do
c(i+1,j+1):=qu(m-i,m,j);

_for k:=1_step 1_until r_do
_for i:=0_step 1_until p-1_do _begin
s:=ss:=0;
_for l:=p_step 1_until q1_do
s:=s+aa(n1+m-1,l,k)×qu(m-i,m,l);
_for l:=m_step -1_until m-i+1_do

```

```

ss:=ss+(aa[n1+l-1,q1,κ]-aa[n1+l-2,q1,κ])×
qu(m-i,m-i,q1);
b[κ,i+1]:=f[n1+m-i-1,κ]-s+ss;
_end _end ;
mvr(c,b,p,r);
_for κ:=1_step 1_until r_do
_for i:=0_step 1_until p-1_do
aa[n1+m-1,i,κ]:=b[κ,i+1]_end ;
_end ;
s1:=xx[n]-xx[n-1];
_for κ:=1_step 1_until r_do
_for j:=0_step 1_until p-1_do _begin
aa[n1-1,j,κ]:=aa[n1,j,κ];
s:=0;
_for i:=q1_step -1_until j_do
s:=s×s1/(i-j+1)+aa[n2-1,i,κ];
aa[n2,j,κ]:=s;
_end ;
_end repres2;

_real _procedure fn2(n1,n2,p,x,aa,κ,x0);
_begin
_integer i,j;_real s,c;
_if x0<x[n1]_then _begin i:=n1-1;_goto lab2_end ;
_if x0>x[n2]_then _begin i:=n2;_goto lab2_end ;
i:=n1;
lab1:_if x[i]≤x0^x[i+1]≥x0_then
_goto lab2_else _begin i:=i+1;_goto lab1_end ;
lab2:s:=0;c:=x0-x[max(i,n1)];
_for j:=2×p_step -1_until 0_do
s:=s×c/(j+1)+aa[i,j,κ];
fn2:=s;

```

```

_end fn2;

_procedure bespar2(p,n1,n2,x,f,eps,al,laδ);
_begin
_integer i,j,k,n;
n:=n2-n1+1;_begin
_real ek,s,t,m,q,φal,φ1al;
_array aa[1:p,1:p],bb[1:1,1:p],h[1:n-p-1,1:p+1],
l,al[1:n-p-1,0:p],φ,φ2[1:n-p-1,1:1],
f1,f2[n1:n2-1,1:1];
t:=(x[n1]+x[n2])/2;m:=x[n2]-x[n1];
_for i:=0_step 1_until p-1_do
_for j:=0_step 1_until p-1_do _begin
s:=0;_for k:=n1_step 1_until n2-1_do
s:=(((x[k+1]-t)/m)^(i+1)-((x[k]-t)/m)^(i+1))×
(((x[k+1]-t)/m)^(j+1)-((x[k]-t)/m)^(j+1))/
(x[k+1]-x[k]))^2+s;
aa[i+1,j+1]:=s/fac(i+1)/fac(j+1)×m×m_end ;
_for i:=0_step 1_until p-1_do _begin
s:=0;_for k:=n1_step 1_until n2-1_do
s:=s+f[k,1]×(((x[k+1]-t)/m)^(i+1)-
((x[k]-t)/m)^(i+1))/(x[k+1]-x[k]);
bb[1,i+1]:=s/fac(i+1)×m_end ;
mvr(aa,bb,p,1);
ek:=0;
_for k:=n1_step 1_until n2-1_do _begin
s:=0;

q:=x[k+1]-x[k];
_for i:=0_step 1_until p-1_do
s:=s+bb[1,i+1]×(((x[k+1]-t)/m)^(i+1)-
((x[k]-t)/m)^(i+1))×m/q/fac(i+1);
ek:=ek+(s-f[k,1])^2;

```

```

_end ;
ek:=sqrt(ek/(n2-n1));
_if eps>ek_then _goto lab;
me
_for i:=n1_step 1_until n2-1_do
f1[i,1]:=f[i,1];
assh2(n1,n2,x,p,h);
assa2(n1,n2,x,p,h,al,a);
decom62(n1,n2,p,a,l);
solve2(n1,n2,p,h,al,1,l,f1,φ);
φal:=φ1al:=0;
_for i:=n1_step 1_until n2-1_do
φal:=(f[i,1]-f1[i,1])2+φal;
φal:=sqrt(φal/(n-1));
_if abs(φal-eps)≤eps/10_then _begin
_for i:=n1_step 1_until n2-1_do
f1[i,1]:=f1[i,1];_goto fin_end ;
_for i:=n1_step 1_until n2-1_do
f2[i,1]:=(f1[i,1]-f[i,1])/al;
solve2(n1,n2,p,h,al,1,l,f2,φ2);

_for i:=n1_step 1_until n2-1_do
f2[i,1]:=f2[i,1]-(f1[i,1]-f[i,1])/al;
_for i:=n1_step 1_until n2-1_do
φ1al:=φ1al+f2[i,1]×(f[i,1]-f1[i,1]);
φ1al:=φal/al-φ1al/n/φal;
al:=al2×φ1al/(al×φ1al-φal+φal2/eps);
_goto me;
fin:_end ;
_end bespar;

```

ОБЩАЯ ЧАСТЬ *ODD* И *EVEN*

```

_array nw[1:10,1:10,1:2];
nw[1,1,1]:=0;nw[1,1,2]:=2;
nw[2,1,1]:=0.577350269189;
nw[2,2,1]:=-nw[2,1,1];
nw[2,1,2]:=nw[2,2,2]:=1;
nw[3,1,1]:=0;
nw[3,2,1]:=0.774596669241;
nw[3,3,1]:=-nw[3,2,1];
nw[3,1,2]:=8/9;
nw[3,2,2]:=nw[3,3,2]:=5/9;
nw[4,1,1]:=0.339981043585;
nw[4,2,1]:=-nw[4,1,1];
nw[4,3,1]:=0.861136311594;
nw[4,4,1]:=-nw[4,3,1];
nw[4,1,2]:=nw[4,2,2]:=0.652145154862;
nw[4,3,2]:=nw[4,4,2]:=0.347854845137;
nw[5,1,1]:=0;
nw[5,2,1]:=0.538469310106;
nw[5,3,1]:=-nw[5,2,1];
nw[5,4,1]:=0.906179845939;
nw[5,5,1]:=-nw[5,4,1];
nw[5,1,2]:=0.568888888889;
nw[5,2,2]:=nw[5,3,2]:=0.478628670499;
nw[5,4,2]:=nw[5,5,2]:=0.236926885056;
nw[6,1,1]:=0.238619186083;
nw[6,2,1]:=-nw[6,1,1];
nw[6,3,1]:=0.661209386466;
nw[6,4,1]:=-nw[6,3,1];
nw[6,5,1]:=0.932469514203;
nw[6,6,1]:=-nw[6,5,1];

```

$nw[6,1,2]:=nw[6,2,2]:=0.467913934573;$
 $nw[6,3,2]:=nw[6,4,2]:=0.360761573048;$
 $nw[6,5,2]:=nw[6,6,2]:=0.171324492379;$
 $nw[7,1,1]:=0;$
 $nw[7,2,1]:=0.405845151377;$
 $nw[7,3,1]:=-nw[7,2,1];$
 $nw[7,4,1]:=0.741531185599;$
 $nw[7,5,1]:=-nw[7,4,1];$
 $nw[7,6,1]:=0.949107912343;$
 $nw[7,7,1]:=-nw[7,6,1];$
 $nw[7,1,2]:=0.417959183673;$
 $nw[7,2,2]:=nw[7,3,2]:=0.381830050505;$
 $nw[7,4,2]:=nw[7,5,2]:=0.279705391489;$
 $nw[7,6,2]:=nw[7,7,2]:=0.129484966169;$
 $nw[8,1,1]:=0.183434642496;$
 $nw[8,2,1]:=-nw[8,1,1];$
 $nw[8,3,1]:=0.525532409916;$
 $nw[8,4,1]:=-nw[8,3,1];$
 $nw[8,5,1]:=0.796666477414;$
 $nw[8,6,1]:=-nw[8,5,1];$
 $nw[8,7,1]:=0.960289856497;$
 $nw[8,8,1]:=-nw[8,7,1];$
 $nw[8,1,2]:=nw[8,2,2]:=0.362683783378;$
 $nw[8,3,2]:=nw[8,4,2]:=0.313706645878;$
 $nw[8,5,2]:=nw[8,6,2]:=0.222381034453;$
 $nw[8,7,2]:=nw[8,8,2]:=0.101228536290;$
 $nw[9,1,1]:=0;$
 $nw[9,2,1]:=0.324253423404;$
 $nw[9,3,1]:=-nw[9,2,1];$
 $nw[9,4,1]:=0.613371432701;$
 $nw[9,5,1]:=-nw[9,4,1];$

```

nw[9,6,1]:=0.836031107327;
nw[9,7,1]:=-nw[9,6,1];
nw[9,8,1]:=0.968160239508;nw[9,9,1]:=-nw[9,8,1];
nw[9,1,2]:=0.330239355001;
nw[9,2,2]:=nw[9,3,2]:=0.312347077040;
nw[9,4,2]:=nw[9,5,2]:=0.260610696403;
nw[9,6,2]:=nw[9,7,2]:=0.180648160695;
nw[9,8,2]:=nw[9,9,2]:=0.081274388362;
nw[10,1,1]:=0.148874338982;
nw[10,2,1]:=-nw[10,1,1];
nw[10,3,1]:=0.433395394129;
nw[10,4,1]:=-nw[10,3,1];
nw[10,5,1]:=0.679409568299;
nw[10,6,1]:=-nw[10,5,1];
nw[10,7,1]:=0.865063366689;
nw[10,8,1]:=-nw[10,7,1];
nw[10,9,1]:=0.973906528517;
nw[10,10,1]:=-nw[10,9,1];
nw[10,1,2]:=nw[10,2,2]:=0.295524224715;
nw[10,3,2]:=nw[10,4,2]:=0.269266719310;
nw[10,5,2]:=nw[10,6,2]:=0.219086362516;
nw[10,7,2]:=nw[10,8,2]:=0.149451349151;
nw[10,9,2]:=nw[10,10,2]:=0.066671344309;
_integer _procedure fac(i);
_begin _integer j;_real a;
_if i=0_then _begin fac:=1;_goto koweu_end ;
a:=1;_for j:=2_step 1_until i_do a:=a*j;
fac:=a;koweu:
_end fac;

```


ТЕКСТ КОМПЛЕКТА *GREEN*

```

_procedure formh(n,m,n1,r0,ii,a0,a,h,j1,ma,mi,arm);
_array a,h;
_integer _array j1;
_label arm;
_integer n,m,n1,ii;
_real r0;
_begin
_integer i,j,k,l,m2,n3,jj,a1;
_integer _array j2(1:n1,1:ii);
_real t;
jj:=n+ii;
ma:=mi:=a0(1,1);
_for i:=1_step 1_until n1_do
_for j:=1_step 1_until n_do
_begin _if a0(j,i)>ma_then ma:=a0(j,i)
_else _if a0(j,i)<mi_then mi:=a0(j,i)_end ;
mi:=mi- $\epsilon$ -5;
ma:=ma-mi;
_for i:=1_step 1_until n1_do
_for j:=1_step 1_until n_do
a1(j,i):=(a0(j,i)-mi)/ma;
_for i:=1_step 1_until n1_do
_for j:=1_step 1_until ii_do
_begin
j2(i,j):=0;
j1(i,j):=0;
_end ;
_for j:=1_step 1_until m_do
_begin
j1(1,j):=j-1;

```

```

j2(l,j):=j1(l,j);
_end ;
n3:=m;
_for i:=1_step 1_until n1-1_do
_begin
m2:=n3;
_for j:=1_step 1_until m2_do
_begin
a1:=0;
_for k:=1_step 1_until n1_do
a1:=a1+j2(k,j);
_if a1≥m-1_then _goto w;
_for k:=n3+1_step 1_until n3+m-a1-1_do
_begin
_for l:=1_step 1_until i_do
j2(l,k):=j2(l,j);
j2(l+1,k):=k-n3;
_end ;
n3:=n3+m-a1-1;
w:
_if n3≥li_then _goto v;
_end ;
v:
_for k:=1_step 1_until ii_do
_begin
_for l:=1_step 1_until n1_do
j1(l,k):=j2(l,k);
_end
_end ;
_for i:=1_step 1_until n_do
_for j:=1_step 1_until ii_do

```

```

_begin
t:=1;
_for k:=1_step 1_until n1_do
t:=t×a[i,k]↑j1[k,j];
h[i,j+n]:=h[j+n,i]:=t
_end ;
_if ((n1/2-entier(n1/2+10-6))>0.1)_then
_begin
a1:=2×m-n1;
_for i:=1_step 1_until n_do
_for j:=i+1_step 1_until n_do
_begin t:=0;
_for k:=1_step 1_until n1_do
t:=t+(a[i,k]-a[j,k])↑2;
h[i,j]:=h[j,i]:=sqrt(t)↑a1
_end
_end
_else
_begin
a1:=m-n1/2;
_for i:=1_step 1_until n_do
_for j:=i+1_step 1_until n_do
_begin t:=0;
_for k:=1_step 1_until n1_do
t:=t+(a[i,k]-a[j,k])↑2;
h[i,j]:=h[j,i]:=ln(t)×t↑a1
_end ;
_end ;
_if r0>0_then t:=(-1)↑entier((n1-1)/2+0.1)/r0_else t:=0;
_for i:=1_step 1_until n_do
h[i,i]:=t;

```

```

_for i:=n+1_step 1_until jj_do
_for j:=n+1_step 1_until jj_do
h(i,j):=0;
Invert2(h,jj, $\epsilon$ -10,alarm);
_goto ff;
alarm:output('t','вырождена','//');
_goto arm;ff:_end

_procedure sol(h,r,jj,n,b);
_array h,r,b;
_integer jj,n;
_begin
_integer i,j;
_for i:=1_step 1_until jj_do
_begin b(i):=0;
_for j:=1_step 1_until n_do
b(i):=b(i)+h(i,j)*r(j)_end _end ;
_real _procedure val(n,m,n1,ii,c1,b,j1,a,ma,mi);
_array c1,b,a;
_integer _array j1;
_integer n,m,n1,ii;
_begin
_integer i,j,f;
_real t;
_real  $\kappa$ ;
_array c(1:n);
_for i:=1_step 1_until n1_do
c(i):=(c1(i)-mi)/ma;
 $\kappa$ :=0;_if (n1/2-entier(n1/2+ $\epsilon$ -b))>0.1_then
_begin
f:=2*m-n1;
_for i:=1_step 1_until n_do

```

```

3 _begin t:=0;
  _for j:=1_step 1_until n1_do
    t:=t+(c[j]-a[i,j])2;
    κ:=κ+b[l]×sqrt(t)f_end _end
  _else _begin f:=m-n1/2;
    _for i:=1_step 1_until n_do
      _begin t:=0;
        _for j:=1_step 1_until n1_do
          t:=t+(c[j]-a[i,j])2;
          _if (t>10-18)_then κ:=κ+b[i]×tf×ln(t)
        _end _end ;
        _for i:=1_step 1_until n1_do
          _begin _if c[i]=0_then c[i]:=10-18_end ;
            _for i:=1_step 1_until ii_do
              _begin t:=b[i+n];
                _for j:=1_step 1_until n1_do
                  t:=t×c[j]j1[j,i];κ:=κ+t_end ;
                val:=κ;
              _end ;
            _procedure besfor(n,m,n1,r0,ii,a0,a,j1,ma,mi,arm,
              eps,d,b1);
            _begin
              _real t,t1;
              _array b[1:ii,1:n],c[1:ii,1:ii],e[1:ii];
              _integer l,j,κ,l,m2,n3;jj,a1;
              _integer _array j2[1:n1,1:ii];
              jj:=n+ii;
              ma:=mi:=a0[1,1];
              _for i:=1_step 1_until n1_do
                _for j:=1_step 1_until n_do
                  _begin _if a0[j,i]>ma_then ma:=a0[j,i]

```

```

_else _if a0(j,i)<mi_then mi:=a0(j,i)_end ;
mi:=mi-10-5;
ma:=ma-mi;
_for i:=1_step 1_until n1_do
_for j:=1_step 1_until n_do
a(j,i):=(a0(j,i)-mi)/ma;
_for i:=1_step 1_until n1_do
_for j:=1_step 1_until ii_do
_begin
j2(i,j):=0;
j1(i,j):=0;
_end ;
_for j:=1_step 1_until m_do
_begin
j1(1,j):=j-1;
j2(1,j):=j1(1,j);
_end ;
n3:=m;
_for i:=1_step 1_until n1-1_do
_begin
m2:=n3;
_for j:=1_step 1_until m2_do
_begin
a1:=0;
_for k:=1_step 1_until n1_do
a1:=a1+j2(k,j);
_if a1≥m-1_then _goto w;
_for k:=n3+1_step 1_until n3+m-a1-1_do
_begin
_for l:=1_step 1_until i_do
j2(l,k):=j2(l,j);

```

```

j2[i+1,k]:=k-n3;
_end ;
n3:=n3+m-a1-1;
w:
_if n3≥ii_then _goto v;
_end ;
v:
_for k:=1_step 1_until ii_do
_begin
_for l:=1_step 1_until n1_do
j1[l,k]:=j2[l,k];
_end
_end ;
m2:=0;
pk:
_for i:=1_step 1_until ii_do
_for j:=1_step 1_until n_do
b[i,j]:=1;
_for i:=1_step 1_until ii_do
_begin
_for j:=1_step 1_until n1_do
_begin
_if j1[j,i]>0_then
_begin
_for k:=1_step 1_until n_do
b[i,k]:=b[i,k]×a[k,j]†j1[j,i]_end _end _end ;
_for i:=1_step 1_until ii_do
_for j:=i_step 1_until ii_do
_begin t:=0;
_for k:=1_step 1_until n_do
t:=b[i,k]×b[j,k]+t;
c[i,j]:=c[j,i]:=t/n_end ;

```

```

_for i:=1_step 1_until ii_do
_begin t:=0;
_for j:=1_step 1_until n_do
t:=t+δ(i,j)×d(j);e(i):=t/n_end ;
_if m2=1_then
_for i:=1_step 1_until ii_do
c(i,i):=c(i,i)+10-3;
invert2(c,ii,10-10,alarm);
_goto ff;
alarm:m2:=1;_goto pk;
ff:_for i:=1_step 1_until ii_do
_begin t:=0;
_for j:=1_step 1_until ii_do
t:=t+c(i,j)×e(j);δ1(i+n):=t_end ;
t1:=0;
_for i:=1_step 1_until n_do
_begin t:=0;
_for j:=1_step 1_until ii_do
t:=t+δ(j,i)×δ1(j+n);
t1:=t1+(t-d(i))2_end ;
t1:=sqrt(t1/n);
_if eps>t1_then _begin
eps:=t1;
output('t','решением является полином');
output('/');
output('t','екр = ');
output('e',t1);
_for i:=1_step 1_until n_do
δ1(i):=0;
output('/', 'e',δ1);
_goto arm_end ;

```



```

_begin
_array h[1:jj,1:jj],q,s[1:n];
_integer i1;
_real rc;
rc:=(-1)entier((n1-1)/2+0.1);
t1:=0;
_for i1:=1_step 1_until 20_do
_begin
_for i:=1_step 1_until n_do
_for j:=1_step 1_until ii_do
h[i,j+n]:=h[j+n,i]:=b[j,i];
_if ((n1/2-entier(n1/2+10-6))>0.1)_then
_begin
a1:=2×m-n1;
_for i:=1_step 1_until n_do
_for j:=i+1_step 1_until n_do
_begin t:=0;
_for k:=1_step 1_until n1_do
t:=t+(a[i,k]-a[j,k])2;
h[i,j]:=h[j,i]:=sqrt(t)a1
_end
_end
_else
_begin
a1:=m-n1/2;
_for i:=1_step 1_until n_do
_for j:=i+1_step 1_until n_do
_begin t:=0;
_for k:=1_step 1_until n1_do
t:=t+(a[i,k]-a[j,k])2;
h[i,j]:=h[j,i]:=ln(t)×ta1

```

```

_end ;
_end ;
t:=rc/r0;
_for i:=1_step 1_until n_do
h[i,i]:=t;
_for i:=n+1_step 1_until jj_do
_for j:=n+1_step 1_until jj_do
h[i,j]:=0;
invert2(h,jj,10-10,alarm);
_goto fp;
alarm:output('t','выпущена','//');
_goto arm;
fp:_for i:=1_step 1_until n_do
_begin q[i]:=0;
_for j:=1_step 1_until n_do
q[i]:=q[i]+h[i,j]*d[j]_end ;
_for i:=1_step 1_until n_do
_begin s[i]:=0;
_for j:=1_step 1_until n_do
s[i]:=s[i]-h[i,j]*q[j]_end ;
t:=t1:=0;
_for i:=1_step 1_until n_do
_begin
t:=t+q[i]*s[i];t1:=t1+q[i]2_end ;
t:=(t1/r0+t/r02)/n;
t1:=sqrt(t1/n)/r0;
_if abs(eps-t1)<eps/10_then _goto lp;
r0:=r0-r02*((t12-t13/eps)/t;
_end ;
lp:
_for i:=1_step 1_until n_do

```

```

b1(i):=q(i);
_for i:=1+n_step 1_until jj_do
_begin b1(i):=0;
_for j:=1_step 1_until n_do
b1(i):=b1(i)+h(i,j)*d(j)_end
_end
_end ;
-procedure formh2(n,n2,n4,m,n1,r0,r1,r2,ii,a0,a2,a3,
a,a4,a6,h,j1,ma,mi,arm);
_begin
-integer i,j,k,l,m2,n3,jj,a1;
-integer k1,k2,k3,ap,aq,af,an,am,ad,d,n7,n8;
-integer _array j2(1:n1,1:ii);
-real t,t1,t2,t3,t4,t5,t6,t7,q,b,c;
_array a7(1:n1);
n8:=n+n2;n7:=n8+n4;jj:=n7+ii;
-if n7<ii_then _goto alarm;
-if ma=0.12345_then _goto gg;
ma:=mi:=a0(1,1);
_for i:=1_step 1_until n1_do
_begin
_for j:=1_step 1_until n_do
_begin _if a0(j,i)>ma_then ma:=a0(j,i)
-else _if a0(j,i)<mi_then mi:=a0(j,i)_end ;
_for j:=1_step 1_until n2_do
_begin _if a2(j,i)>ma_then ma:=a2(j,i)
-else _if a2(j,i)<mi_then mi:=a2(j,i)_end ;
_for j:=1_step 1_until n4_do
_begin _if a3(j,i)>ma_then ma:=a3(j,i)
-else _if a3(j,i)<mi_then mi:=a3(j,i)_end ;
_end ;

```

```

mi:=mi-10-5;
ma:=ma-mi;
_for i:=1_step 1_until n1_do
_begin
_for j:=1_step 1_until n_do
a(j,i):=(a0(j,i)-mi)/ma;
_for j:=1_step 1_until n2_do
_begin
a4(j,i):=(a2(j,i)-mi)/ma;
a4(j,i+n1):=a2(j,i+n1)_end ;
_for j:=1_step 1_until n4_do
a6(j,i):=(a3(j,i)-mi)/ma_end ;
_for i:=1_step 1_until n4_do
_begin k2:=0;
_for j:=1_step 1_until n1_do
_begin k2:=k2+n1+2-j;
a6(i,k2):=a3(i,j+n1)2;
_for k:=j+1_step 1_until n1_do
a6(i,k2+k-j):=2×a3(i,j+n1)×a3(i,k+n1)_end _end ;
_for i:=1_step 1_until n1_do
_for j:=1_step 1_until ii_do
_begin
j2(i,j):=0;
j1(i,j):=0;
_end ;
_for j:=1_step 1_until m_do
_begin
j1(1,j):=j-1;
j2(1,j):=j1(1,j);
_end ;
n3:=m;

```

```

_for i:=1_step 1_until n1-1_do
_begin
m2:=n3;
_for j:=1_step 1_until m2_do
_begin
a1:=0;
_for k:=1_step 1_until n1_do
a1:=a1+j2(k,j);
_if a1>=m-1_then _goto w;
_for k:=n3+1_step 1_until n3+m-a1-1_do
_begin
_for l:=1_step 1_until i_do
j2(l,k):=j2(l,j);
j2(l+1,k):=k-n3;
_end ;
n3:=n3+m-a1-1;
w:
_if n3>=11_then _goto v;
_end ;
v:
_for k:=1_step 1_until 11_do
_begin
_for l:=1_step 1_until n1_do
j1(l,k):=j2(l,k);
_end
_end ;
gg:
_if (n1/2-entier(n1/2+10-6))>0.1_then d:=1
_else d:=0;
_for i:=1_step 1_until n_do
_for j:=1_step 1_until ii_do

```

```

_begin
t:=1;
_for k:=1_step 1_until n1_do
t:=t*a1[i,k]*j1[k,j];
h[i,j+n7]:=h[j+n7,i]:=t
_end ;
_for i:=1_step 1_until n2_do
_for j:=1_step 1_until ii_do
_begin t:=0;
t1:=1;
_for k1:=1_step 1_until n1_do
_begin
t1:=t1*a4[i,k1]*j1[k1,j];
t:=t+(a4[i,k1+n1]*j1[k1,j])/a4[i,k1]_end ;
t:=t*t1;
h[n+i,n7+j]:=h[n7+j,n+i]:=t_end ;
_for i:=1_step 1_until n4_do
_for j:=1_step 1_until ii_do
_begin
t:=t2:=0;n3:=0;
t3:=1;
_for k:=1_step 1_until n1_do
t3:=t3*a6[i,k]*j1[k,j];
_for k:=1_step 1_until n1_do
_begin n3:=n3+n1-k+2;_if j1[k,j]>0_then
_begin j1[k,j]:=j1[k,j]-1;
_for k1:=k_step 1_until n1_do
t2:=t2+(a6[i,k1-k+n3]*j1[k1,j])/a6[i,k1];
j1[k,j]:=j1[k,j]+1;t:=t+t2*t3*j1[k,j]/a6[i,k];t2:=0_end
_end ;
h[n8+i,n7+j]:=h[n7+j,n8+i]:=t_end ;

```

```

_if d=1_then a1:=2×m-n1_else a1:=m-n1/2;
_for i:=1_step 1_until n_do
_for j:=i+1_step 1_until n_do
_begin t:=0;
_for k:=1_step 1_until n1_do
t:=t+(a[i,k]-a[j,k])2;
_if d=1_then t1:=sqrt(t)a1_else t1:=ln(t)×ta1;
h[i,j]:=h[j,i]:=t1
_end ;
ap:=a1;_if d=1_then a1:=a1-2_else a1:=a1-1;
_for i:=1_step 1_until n_do
_for j:=1_step 1_until n2_do
_begin t:=t1:=0;
_for k:=1_step 1_until n1_do
_begin a7[k]:=a[i,k]-a4[j,k];t:=t+a7[k]2_end ;
_if t>10-18_then _begin
_for k:=1_step 1_until n1_do
t1:=t1+a4[j,k+n1]×a7[k];
_if d=1_then t1:=t1×ap×sqrt(t)a1_else
t1:=t1×2×(ap×ln(t)+1)×ta1;
_end ;
h[i,n+j]:=h[n+j,i]:=-t1_end ;
_if d=1_then _begin aq:=a1×ap;ap:=a1;a1:=a1-2_end
_else _begin aq:=a1;a1:=a1-1_end ;
_for i:=1_step 1_until n_do
_for j:=1_step 1_until n4_do
_begin
t:=t1:=t2:=0;n3:=0;
_for k:=1_step 1_until n1_do
_begin a7[k]:=a[i,k]-a6[j,k];t:=t+a7[k]2_end ;
_if t>10-12_then _begin

```

```

_for k:=1_step 1_until n1_do
_begin n3:=n3+n1-k+2;
_for k1:=k_step 1_until n1_do
t1:=t1+a6(j,n3+k1-k)*a7(k)*a7(k1);
_end ;
_if d=1_then t1:=(t/ap+t1)*sqrt(t)*a1*aq_else
_begin t3:=ap*ln(t)+1;t1:=t*a1*(t3*t*2+t1*4*
(aq*t3+ap))_end ;
_end ;
h(i,n8+j):=h(n8+j,i):=t1_end ;
_for i:=1_step 1_until n2_do
_for j:=i+1_step 1_until n2_do
_begin t:=t1:=t2:=0;
_for k:=1_step 1_until n1_do
_begin a7(k):=a4(i,k)-a4(j,k);t:=t+a7(k)*t2_end ;
_if t≠0_then _begin
_for k:=1_step 1_until n1_do
_for k1:=1_step 1_until n1_do
t1:=t1+a4(i,k+n1)*a4(j,k1+n1)*a7(k)*a7(k1);
_for k:=1_step 1_until n1_do
t2:=t2+a4(i,k+n1)*a4(j,k+n1);
_if d=1_then t1:=(t2*t/ap+t1)*sqrt(t)*a1*aq_else
_begin t3:=ap*ln(t)+1;t1:=t*a1*(t2*t3*t*2+t1*4*
(aq*t3+ap))_end _end ;
h(n+i,n+j):=h(n+j,n+i):=-t1_end ;
_if d=1_then _begin
aq:=aq*a1;ap:=a1;a1:=a1-2;
_end _else _begin
af:=a1;a1:=a1-1;
am:=ap*aq_end ;
_for i:=1_step 1_until n2_do

```



```

_for j:=1_step 1_until n4_do
_begin
t:=t1:=t2:=t3:=0;n3:=0;
t4:=t5:=0;
_for k:=1_step 1_until n1_do
_begin a7[k]:=a4[i,k]-a6[j,k];t:=t+a7[k]2_end ;
_if t≠0_then _begin
_for k2:=1_step 1_until n1_do
_begin b:=a7[k2]2;
_for k1:=1_step 1_until k2-1_do
_begin n3:=n3+n1-k1+2;q:=a7[k2]×a7[k1];
t3:=t3+(a6[j,n3]×a7[k2]+a6[j,n3+k2-k1]×a7[k1]);
_for k:=k1_step 1_until n1_do
t1:=t1+a6[j,k-k1+n3]×q×a7[k]_end ;
n3:=n3+n1-k2+2;
t3:=t3+a6[j,n3]×a7[k2]×2;
_for k1:=k2_step 1_until n1_do
_begin t2:=a6[j,n3+k1-k2]×a7[k1];t1:=t1+t2×b;
t3:=t3+t2_end ;
_for k1:=k2+1_step 1_until n1_do
_begin n3:=n3+n1-k1+2;q:=a7[k2]×a7[k1];
_for k:=k1_step 1_until n1_do
t1:=t1+a6[j,n3+k-k1]×q×a7[k];
t3:=t3+a6[j,n3]×a7[k2]_end ;
t4:=t4+t1×a4[i,n1+k2];t5:=t5+t3×a4[i,n1+k2];
t1:=t3:=0;n3:=0;
_end ;
_if d=1_then t2:=(t4+t5×t/ap)×sqrt(t)a1×aq_else
_begin t3:=(ap×ln(t)+1)×aq+ap;
t2:=ta1×(4×t5×t3×t+8×t4×(af×t3+am))_end _end ;
h[n+i,n8+j]:=h[n8+j,n+i]:=t2_end ;

```

```

_if d=1_then _begin
af:=aq×a1;a1:=a1-2;ap:=aq/ap;
_end _else _begin an:=a1;a1:=a1-1;ad:=am×af_end ;
_for i:=1_step 1_until n4_do
_for j:=i+1_step 1_until n4_do
_begin
t:=t1:=t2:=t3:=t5:=t6:=t7:=0;
m2:=0;
_for κ:=1_step 1_until n1_do
_begin a7[κ]:=a6[i,κ]-a6[j,κ];t:=t+a7[κ]↑2_end
_if t≠0_then _begin
_for κ3:=1_step 1_until n1_do
_begin m2:=m2+n1-κ3+2;q:=a7[κ3]↑2;
n3:=0;
_for κ1:=1_step 1_until κ3-1_do
_begin n3:=n3+n1-κ1+2;
_for κ:=κ1_step 1_until n1_do
_begin t4:=a6[j,n3+κ-κ1]×a7[κ1]×a7[κ];
t1:=t1+t4×q;t2:=t2+t4_end ;
t2:=t2+a6[j,n3]×q+a6[j,n3+κ3-κ1]×a7[κ3]×a7[κ1]×2;
t3:=t3+a6[j,n3]_end ;
n3:=n3+n1-κ3+2;
_for κ1:=κ3_step 1_until n1_do
_begin t4:=a6[j,n3+κ1-κ3]×a7[κ3]×a7[κ1];
t2:=t2+3×t4;t1:=t1+t4×q_end ;
t4:=a6[j,n3]×3;t3:=t3+t4;t2:=t2+t4×q;
_for κ1:=κ3+1_step 1_until n1_do
_begin n3:=n3+n1-κ1+2;
_for κ:=κ1_step 1_until n1_do
_begin t4:=a6[j,n3+κ-κ1]×a7[κ1]×a7[κ];
t1:=t1+t4×q;t2:=t2+t4_end ;

```

```

t3:=t3+a6[j,n3];t2:=t2+q*a6[j,n3]_end);
t5:=t5+t1*a6[i,m2];t6:=t6+t2*a6[i,m2];
t7:=t7+t3*a6[i,m2];t1:=t2:=t3:=0;
/_for k2:=k3+1_step 1_until n1_do
  _begin n3:=0;b:=a7[k3]*a7[k2];
  _for k1:=1_step 1_until k3-1_do
    _begin n3:=n3+n1-k1+2;
    _for k:=k1_step 1_until n1_do
      t1:=t1+a6[j,n3+k-k1]*a7[k]*a7[k1]*b;
      t2:=t2+a6[j,n3]*b+a6[j,n3+k3-k1]*a7[k2]*a7[k1];
      t2:=t2+a6[j,n3+k2-k1]*a7[k3]*a7[k1]_end ;
      n3:=n3+n1-k3+2;
      _for k:=k3_step 1_until n1_do
        _begin t4:=a6[j,n3+k-k3]*a7[k2]*a7[k];t2:=t2+t4;
        t1:=t1+t4*q_end ;
        t2:=t2+2*a6[j,n3]*b+a6[j,n3+k2-k3]*q;
        t7:=t7+a6[j,n3+k2-k3]*a6[i,m2+k2-k3];
        _for k1:=k3+1_step 1_until k2-1_do
          _begin n3:=n3+n1-k1+2;c:=b*a7[k1];
          _for k:=k1_step 1_until n1_do
            t1:=t1+a6[j,n3+k-k1]*c*a7[k];
            t2:=t2+a6[j,n3]*b+a6[j,n3+k2-k1]*a7[k3]*a7[k1]_end
            n3:=n3+n1-k2+2;c:=a7[k2]*t2;
            _for k:=k2_step 1_until n1_do
              _begin t4:=a6[j,n3+k-k2]*a7[k3]*a7[k];
              t1:=t1+t4*c;t2:=t2+t4_end ;
              t2:=t2+a6[j,n3]*b*2;
              _for k1:=k2+1_step 1_until n1_do
                _begin n3:=n3+n1-k1+2;c:=b*a7[k1];
                _for k:=k1_step 1_until n1_do
                  t1:=t1+a6[j,n3+k-k1]*c*a7[k];

```

```

t2:=t2+a6(j,n3)*b_end ;c:=a6(i,n2+κ2-κ3);
t5:=t5+t1*c;t6:=t6+t2*c;t1:=t2:=0_end_end ;
_if d=1_then t3:=sqrt(t)*a1*(t5*af+t6*t*aq+t7*ap*t*2)
_else _begin
t1:=aq*(ap*ln(t)+1)+ap;t2:=af*t1+am;
t4:=an*t2+ad;
t3:=t5*16*t4*t*af+t6*8*t2*t*an+t7*4*t1*t*af;
_end_end ;
h(n8+i,n8+j):=h(n8+j,n8+i):=t3_end ;
_if r0>0_then t:=(-1)fentier((n1-1)/2+0.1)/r0_else t:=0;
_for i:=1_step 1_until n_do h(i,i):=t;
_if r1>0_then t:=(-1)fentier((n1-1)/2+0.1)/r1_else t:=0;
_for i:=1_step 1_until n2_do h(n+i,n+1):=t;
_if r2>0_then t:=(-1)fentier((n1-1)/2+0.1)/r2_else t:=0;
_for i:=1_step 1_until n4_do h(n8+i,n8+i):=t;
_for i:=n7+1_step 1_until jj_do
_for j:=n7+1_step 1_until jj_do
h(i,j):=0;
invert2(h,jj,10,alarm);
_goto ff;
alarm:output('t','вырождена','//');
_goto arm;
ff:_end ;
_procedure sol2(h,r,n,n2,n4,jj,ma,b);
_begin
_integer i,j,n7;
_real κ;
n7:=n+n2+n4;
_begin _array rκ(1:n7);
_for i:=1_step 1_until n_do
rκ(i):=r(i);

```

```

_for i:=1_step 1_until n2_do
rk(n+i):=r(n+i)*ma;
κ:=maf2;
_for i:=1_step 1_until n4_do
rk(n+n2+i):=r(n+n2+i)*κ;
_for i:=1_step 1_until jj_do
_begin b(i):=0;
_for j:=1_step 1_until n7_do
b(i):=b(i)+h(i,j)*rk[j]_end _end _end ;

_real _procedure val2(n,n2,n4,m,n1,ii,c1,b,j1,a,a4,a6,ma
,mi);
_array c1,b,a;
_integer _array j1;
_integer n,m,n1,ii;
_begin
_real κ;
_array c(1:n);
_integer i,j,f,g;
_integer d,e,n5,n7,l,r;
_real t,t1,t2,t3;
_array p(1:n1);
n5:=n+n2;n7:=n5+n4;
_for i:=1_step 1_until n1_do
c(i):=(c1(i)-mi)/ma;
κ:=0;_if (n1/2-entier(n1/2+_10^-6))>0.1_then
d:=1_else d:=0;
_if d=1_then f:=2*m-n1_else f:=m-n1/2;
_for i:=1_step 1_until n_do
_begin t:=0;
_for j:=1_step 1_until n1_do
t:=t+(c[j]-a(i,j))↑2;

```

```

_if d≠1_then κ:=κ+δ[i]×sqrt(t)↑f_else
_if (t>10-18)_then κ:=κ+δ[i]×t↑f×ln(t)
_end ;

_if d=1_then e:=f-2_else e:=f-1;
_for i:=1_step 1_until n2_do
_begin t:=t1:=0;
_for j:=1_step 1_until n1_do
_begin p[j]:=c[j]-a4[i,j];t:=t+p[j]↑2_end ;
_if t>10-18_then _begin
_for j:=1_step 1_until n1_do
t1:=t1+a4[i,n1+j]×p[j];
_if d=1_then t3:=f×t1×sqrt(t)↑e_else
t3:=t1×2×(f×ln(t)+1)×t↑e;
κ:=κ-δ[i+n]×t3_end _end ;
_if d=1_then _begin g:=f×e;e:=e-2_end
_else g:=e-1;
_for i:=1_step 1_until n4_do
_begin
t:=t1:=t2:=0;r:=0;
_for j:=1_step 1_until n1_do
_begin p[j]:=c[j]-a6[i,j];t:=t+p[j]↑2_end ;
_if t>10-18_then _begin
_for j:=1_step 1_until n1_do
_begin r:=r+n1-j+2;
_for l:=j_step 1_until n1_do
t2:=t2+a6[i,r+l-j]×p[j]×p[l]_end ;
_if d=1_then t3:=sqrt(t)↑ex(g×t2+f×t)_else
_begin t3:=f×ln(t)+1;t3:=t↑g×(2×t×t3+4×t2×(ex×t3+f))
_end ;
κ:=κ+δ[i+n5]×t3_end _end ;
_for i:=1_step 1_until n1_do

```

```

_begin _if c(i)=0_then c(i):=-10-18_end ;
_for i:=1_step 1_until ii_do
_begin t:=b(i+n7);
_for j:=1_step 1_until n1_do
t:=txc(j)tj1(j,i);k:=k+t_end ;
val2:=k;
_end ;

-procedure besfo2(n,n2,n4,m,n1,r0,r1,r2,ii,a0,a2,a3,
a,a4,a6,j1,ma,mi,arm,eps,d,b1);
_begin
_real t,t1,t2,t3,e0,e2,e1;
_array b(1:ii,1:n),c(1:ii,1:ii),e(1:ii);
_integer i,j,k,l,m2,n3,jj,a1;
_integer _array j2(1:n1,1:ii);
_integer k1,k2,n7,n8;
_array b2(1:ii,1:n2),b4(1:ii,1:n4);
n8:=n+n2;n7:=n8+n4;jj:=n7+ii;
e0:=nxr0;e1:=n2xr1;e2:=n4xr2;
_if n7<ii_then _goto alarm;
ma:=mi:=a0(1,1);
_for i:=1_step 1_until n1_do
_begin
_for j:=1_step 1_until n_do
_begin _if a0(j,i)>ma_then ma:=a0(j,i)
_else _if a0(j,i)<mi_then mi:=a0(j,i)_end ;
_for j:=1_step 1_until n2_do
_begin _if a2(j,i)>ma_then ma:=a2(j,i)
_else _if a2(j,i)<mi_then mi:=a2(j,i)_end ;
_for j:=1_step 1_until n4_do
_begin _if a3(j,i)>ma_then ma:=a3(j,i)
_else _if a3(j,i)<mi_then mi:=a3(j,i)_end ;

```

```

_end ;
mi:=mi-10-5;
ma:=ma-m1;
_for i:=1_step 1_until n1_do
_begin
_for j:=1_step 1_until n_do
a[j,i]:=(a0[j,i]-mi)/ma;
_for j:=1_step 1_until n2_do
_begin
a4[j,i]:=(a2[j,i]-mi)/ma;
a4[j,i+n1]:=a2[j,i+n1]_end ,
_for j:=1_step 1_until n4_do
a6[j,i]:=(a3[j,i]-mi)/ma_end ;
_for i:=1_step 1_until n4_do
_begin k2:=0;
_for j:=1_step 1_until n1_do
_begin k2:=k2+n1+2-j;
a6[i,k2]:=a3[i,j+n1]2;
_for k:=j+1_step 1_until n1_do
a6[i,k2+k-j]:=2×a3[i,j+n1]×a3[i,k+n1]_end _end ;
_for i:=1_step 1_until n1_do
_for j:=1_step 1_until ii_do
_begin
j2[i,j]:=0;
j1[i,j]:=0;
_end ;
_for j:=1_step 1_until m_do
_begin
j1[1,j]:=j-1;
j2[1,j]:=j1[1,j];
_end ;

```



```

n3:=m;
_for i:=1_step 1_until n1-1_do
_begin
m2:=n3;
_for j:=1_step 1_until m2_do
_begin
a1:=0;
_for k:=1_step 1_until n1_do
a1:=a1+j2[k,j];
_if a1≥m-1_then _goto w;
_for k:=n3+1_step 1_until n3+m-a1-1_do
_begin
_for l:=1_step 1_until l_do
j2[l,k]:=j2[l,j];
j2[l+1,k]:=k-n3;
_end ;
n3:=n3+m-a1-1;
w:
_if n3≥ii_then _goto v;
_end ;
v:
_for k:=1_step 1_until ii_do
_begin
_for l:=1_step 1_until n1_do
j1[l,k]:=j2[l,k];
_end
_end ;
m2:=0;pκ:
_for i:=1_step 1_until ii_do
_for j:=1_step 1_until n_do
b[i,j]:=1;

```

```

_for i:=1_step 1_until ii_do
_begin
_for j:=1_step 1_until n1_do
_begin
_if j1[j,i]>0_then
_begin
_for k:=1_step 1_until n_do
b[i,k]:=b[i,k]*a[k,j]*j1[j,i]_end _end _end ;
_for i:=1_step 1_until n2_do
_for j:=1_step 1_until ii_do
_begin t:=0;
t1:=1;
_for k1:=1_step 1_until n1_do
_begin
t1:=t1*a4[i,k1]*j1[k1,j];
t:=t+(a4[i,k1+n1]*j1[k1,j])/a4[i,k1]_end ;
t:=t*t1;b2[j,i]:=t_end ;
_for i:=1_step 1_until n4_do
_for j:=1_step 1_until ii_do
_begin
t:=t2:=0;n3:=0;t3:=1;
_for k:=1_step 1_until n1_do
t3:=t3*a6[i,k]*j1[k,j];
_for k:=1_step 1_until n1_do
_begin n3:=n3+n1-k+2;_if j1[k,j]>0_then
_begin j1[k,j]:=j1[k,j]-1;
_for k1:=k_step 1_until n1_do
t2:=t2+(a6[i,k1-k+n3]*j1[k1,j])/a6[i,k1];
j1[k,j]:=j1[k,j]+1;t:=t+t2*t3*j1[k,j]/a6[i,k];t2:=0_end
_end ;
b4[j,i]:=t_end ;

```

```

_for i:=1_step 1_until ii_do
_begin t:=t1:=t2:=0;
_if r0>0_then _begin
_for k:=1_step 1_until n_do
t:=t+d(k)*b1(i,k);e(i):=t/e0_end
_else e(i):=0;
_if r1>0_then _begin
_for k:=1_step 1_until n2_do
_begin
t1:=t1+d(n+k)*b2(i,k)_end ;
e(i):=e(i)+t1/e1_end ;
_if r2>0_then _begin
_for k:=1_step 1_until n4_do
_begin
t2:=t2+d(n8+k)*b4(i,k)_end ;
e(i):=e(i)+t2/e2_end ;
_if r0>0_then _begin
_for j:=1_step 1_until ii_do
_begin t:=0;
_for k:=1_step 1_until n_do
t:=t+b(i,k)*b(j,k);c(i,j):=t/e0
_end
_end
_else
_for j:=1_step 1_until ii_do
c(i,j):=0;
_if r1>0_then _begin
_for j:=1_step 1_until ii_do
_begin t:=0;
_for k:=1_step 1_until n2_do
t:=t+b2(i,k)*b2(j,k);c(i,j):=c(i,j)+t/e1

```

```

_end _end ;
_if r2>0_then _begin
_for j:=1_step 1_until ii_do
_begin t:=0;
_for k:=1_step 1_until n4_do
t:=t+b4[i,k]×b4[j,k];c[i,j]:=c[i,j]+t/e2_end
_end _end ;
_for i:=1_step 1_until ii_do
_for j:=i+1_step 1_until ii_do
c[j,i]:=c[i,j];
_if m2=1_then
_for i:=1_step 1_until ii_do
c[i,i]:=c[i,i]+10-3;
invert2(c,ii,10-10,alarm);
_goto ff;
alarm:m2:=1;_goto pk;
ff:_for i:=1_step 1_until ii_do
_begin t:=0;
_for j:=1_step 1_until ii_do
t:=t+c[i,j]×e[j];b1[i+n7]:=t_end ;
t1:=0;t2:=t3:=0;
_if r0>0_then _begin
_for i:=1_step 1_until n_do
_begin t:=0;
_for j:=1_step 1_until ii_do
t:=t+b[j,i]×b1[j+n7];
t1:=t1+(t-d[i])2_end ;

t1:=t1/e0_end ;
_if r1>0_then _begin
_for i:=1_step 1_until n2_do
_begin t:=0;

```

```

_for j:=1_step 1_until ii_do
t:=t+b2[j,i]×b1[j+n7];
t2:=t2+(t-d[n+i])↑2_end ;
t1:=t1+t2/e1_end ;
_if r2>0_then _begin
_for i:=1_step 1_until n4_do
_begin t:=0;
_for j:=1_step 1_until ii_do
t:=t+b4[j,i]×b1[j+n7];
t3:=t3+(t-d[n8+i])↑2_end ;
t1:=t1+t3/e2_end ;
t1:=sqrt(t1);
_if eps>t1_then _begin
eps:=t1;
output('t','решением является полином','/');
output('t','exp = ','e',t1);
_for i:=1_step 1_until n7_do b1[i]:=0;
output('/', 'e',b1);
_goto arm_end ;
_begin
_real r,t4,t5;
_array h[1:jj,1:jj],q,s[1:n7];
_integer i1;
_real m6;
m6:=0.12345;r:=1;
_for i1:=1_step 1_until 20_do
_begin
formh2(n,n2,n4,m,n1,r0,r1,r2,il,
a0,a2,a3,a,a4,a6,h,j1,m6,mi,arm);
_for i:=1_step 1_until n7_do
_begin q[i]:=0;

```

```

_for j:=1_step 1_until n7_do
q[i]:=q[i]+h[i,j]*d[j]_end ;
_for i:=1_step 1_until n7_do
_begin s[i]:=0;
_for j:=1_step 1_until n7_do
s[i]:=s[i]-h[i,j]*q[j]_end ;
t:=t1:=0;
_if r0>0_then _begin
_for i:=1_step 1_until n_do
_begin
t:=t+q[i]*s[i];t1:=t1+q[i]2_end ;
t1:=t1/e0;t:=t/e0_end ;
_if r1>0_then _begin
t2:=t3:=0;
_for i:=n+1_step 1_until n8_do
_begin
t2:=t2+q[i]*s[i];t3:=t3+q[i]2_end ;
t1:=t1+t3/e1;t:=t+t2/e1_end ;
_if r2>0_then _begin
t4:=t5:=0;
_for i:=n8+1_step 1_until n7_do
_begin
t4:=t4+q[i]*s[i];t5:=t5+q[i]2_end ;
t1:=t1+t5/e2;t:=t+t4/e2_end ;
t2:=t1;t1:=sqrt(t1)/r;
_if abs(eps-t1)<eps/10_then _goto lp;
t:=t2/r+t/r2;
r:=r-r2*t12*(1-t1/eps)/t;
r0:=r0*r;r1:=r1*r;r2:=r2*r;
_end ;
lp:_for i:=1_step 1_until n7_do

```

```

b1[i]:=q[i];
_for i:=1+n7_step 1_until jj_do
_begin b1[i]:=0;
_for j:=1_step 1_until n7_do
b1[i]:=b1[i]+h[i,j]*d[j]_end
_end
_end ;

```

Приложение 5

ТЕКСТ КОМПЛЕКТА *AVERAGE*

```
_procedure aver(f,n0,nt,p,w,a1,fcl);
_begin
_integer i,j,k,l;_real a,b,c;
k:=entier(p/2+_10^-6);
a:=p/2;
_if abs(k-a)≤_10^-6_then _goto even_else _goto odd;
odd:
k:=entier(w/2+_10^-6);
a:=w/2;
_if abs(k-a)≤_10^-6_then _goto even;
k:=n0;
a:=0;_for i:=1_step 1_until p_do _begin
l:=k-w×(p/2-i+1)+0.5;
b:=0;_for j:=l_step 1_until l+w-1_do
b:=b+f(j);
a:=a+al[i]×b_end ;
fcl[k]:=a/w;
_for k:=n0_step 1_until nt-1_do _begin
a:=0;_for i:=1_step 1_until p_do _begin
j:=k-w×(p/2-i+1)+0.5;
l:=j+w;
a:=a+al[i]×(f(l)-f(j))_end ;
fcl[k+1]:=fcl[k]+a/w_end ;
_goto fin;
even:
k:=n0;
a:=0;_for i:=1_step 1_until p_do _begin
l:=k-w×(p/2-i+1);
b:=0;_for j:=l_step 1_until l+w-1_do
b:=b+f(j)+f(j+1);
```



```

a:=a+a1(i)*b_end ;
fcl(k):=a/w/2;
_for k:=n0_step 1_until nt-1_do _begin
a:=0;_for i:=1_step 1_until p_do _begin
j:=k-w*(p/2-i+1);
l:=j+w;
a:=a+a1(i)*(-f(j+1)-f(j)+f(l+1)+f(l))_end ;
fcl(k+1):=fcl(k)+a/w/2_end ;
fin:
_end aver;

_procedure combi(z,n,a1,p);
_begin _array a[1:p,1:p],b[1:1,1:p];
__integer _array rr[1:2,0:n];
_integer i,i1,j,j1,k,m1,p1;_real r,q,t,b1,b2,pi;
_real _procedure zn1(d,b,m,c1,c2);
_integer m;_real d,b,c1,c2;
_begin _real bM,dM;
_if m=0_then _begin bM:=dM:=1_end
_else _begin bM:=b↑m;dM:=d↑m_end ;
zn1:=(exp(-c1*b)*(c2*sin(c2*b)-c1*cos(c2*b))*bM
-exp(-c1*d)*(c2*sin(c2*d)-c1*cos(c2*d))*dM)/
(c1↑2+c2↑2);_end ;
_real _procedure zn2(d,b,m,c1,c2);
_integer m;_real d,b,c1,c2;
_begin _real bM,dM;
_if m=0_then _begin bM:=dM:=1_end
_else _begin bM:=b↑m;dM:=d↑m_end ;
zn2:=-(exp(-c1*b)*(c1*sin(c2*b)+c2*cos(c2*b))*bM
-exp(-c1*d)*(c1*sin(c2*d)+c2*cos(c2*d))*dM)/
(c1↑2+c2↑2);
_end ;pi:=3.1415926;

```

```

p1:=0;rr[1,0]:=1;rr[2,0]:=0;
_for i:=1_step 1_until n_do
_begin rr[1,i]:=rr[1,i-1]+rr[2,i-1];
_if z[3,i]=0_then _begin rr[2,i]:=
z[1,i]+1;p1:=p1+rr[2,i]_end
_else _begin rr[2,i]:=2*(z[1,i]+1);p1:=p1+rr[2,i]_end
_end ;
_for i:=1_step 1_until n_do
_for j:=i+1_step 1_until n_do
_if abs(z[2,i]-z[2,j])<10-8
_then _begin _if abs((z[3,i]-z[3,j])/(2*pi)
-entier((z[3,i]-z[3,j])/(2*pi)))<10-8
_then _goto error;_end ;
_if p≠p1_then _goto err;
_for i1:=1_step 1_until p_do b[1,i1]:=0;
_for i1:=1_step 1_until n_do
_begin j:=rr[1,i1];r:=z[2,i1];q:=z[3,i1];t:=r↑2+q↑2;
_if z[4,i1]=1_then b[1,j]:=1;
_if z[3,i1]=0_then
_begin
_for κ:=1_step 1_until p_do
_begin b1:=p/2-κ;b2:=b1+1;
_if r=0_then a[j,κ]:=1_else
a[j,κ]:=(exp(-r×b1)-exp(-r×b2))/r;
_for j1:=j+1_step 1_until j+rr[2,i1]-1_do
_begin _if r=0_then a[j1,κ]:=b2↑(j1-j+1)-
b1↑(j1-j+1)_else
a[j1,κ]:=((j1-j)×a[j1-1,κ]-exp(-r×b2)×b2↑(j1-j)
+exp(-r×b1)×b1↑(j1-j))/r_end
_end _end
_else _begin _for κ:=1_step 1_until p_do

```

```

_begin b1:=p/2-k;b2:=b1+1;
a[j,k]:=zn1(b1,b2,0,r,q);
a[j+1,k]:=zn2(b1,b2,0,r,q);
_for j1:=j+2_step 2_until j+rr(2,l1)-1_do
_begin m1:=entier((j1-j)/2+0.1);
a[j1,k]:=zn1(b1,b2,m1,r,q)+m1*
(r*a[j1-2,k]-q*a[j1-1,k])/t;
a[j1+1,k]:=zn2(b1,b2,m1,r,q)+m1*
(q*a[j1-2,k]+r*a[j1-1,k])/t;
_end _end _end ;_end ;
mvr(a,b,p,1);
_for i:=1_step 1_until p_do
all(i):=b(1,i);
_goto en;err:
output('2/');
output('t','параметр p задан неверно');
output('/');
output('задайте p=');output('e',p1,'2/');
_goto en;
error:output('2/');
output('t','система квазимономов d-зависима');
output('2/');
en:_end combi;

_procedure indic(p,a,b,ge,im);
_begin _array c,s,e[1:p];
_integer k;
_real cb,sb,cp,sp,ea,ep,r,re1,im1,re2,im2;
_real _procedure com(a1,b1,c1,d1,er,mi);
_begin er:=a1*c1-b1*d1;mi:=a1*d1+b1*c1;_end ;
r:=a2+b2;
_if r=0_then _begin im:=0;re:=0;

```

```

_for k:=1_step 1_until p_do re:=re+al(k);
_goto m;_end ;
ea:=exp(a);ep:=exp(a*p/2);
cb:=cos(b)*ea-1;sb:=sin(b)*ea;
cp:=cos(b*p/2)*ep/r;sp:=sin(b*p/2)*ep/r;
_for k:=1_step 1_until p_do
_begin e(k):=exp(-a*k);
c(k):=al(k)*e(k)*cos(b*k);
s(k):=-al(k)*e(k)*sin(b*k);_end;
com(a,-b,cp,sp,re1,im1);
com(re1,im1,cb,sb,re2,im2);
re:=im:=0;
_for k:=1_step 1_until p_do
_begin com(re2,im2,c(k),s(k),re1,im1);
re:=re+re1;im:=im+im1_end ;
m:_end .indic;

```

Приложение 6

Таблицы к § 6.1

Таблица 6.1.1

i	0	1	2	3	4	5	6	7	8	9	10
x_i	0	0.8	1.2	1.9	3.0	5.0	7.0	8.1	8.8	9.2	10.0
$\sigma_1(x_i)$	0.0	-0.1	-0.5	1.5	2.0	3.0	2.0	1.5	-0.5	-0.1	0.0
$\sigma_2(x_i)$	-5.0	-4.5	-4.0	-3.5	-4.0	0.0	4.0	3.5	4.0	4.5	5.0

Таблица 6.1.2

j	i	1	2	3	4	5	6	7	8
1	1	-0.5482	-1.0331	-0.2089	-0.0575	-0.0245	-0.0424	-0.2296	-0.6835
2	2	2.8409	1.9841	0.4189	0.1136	0.0806	0.2525	1.1806	2.9762
3	3	-2.9762	-1.1806	-0.2525	-0.0806	-0.1136	-0.4189	-1.9841	-2.8409
4	4	0.6835	0.2296	0.0424	0.0245	0.0575	0.0089	1.0331	0.5482

Таблица 6.1.3

j	i	1	2	3	4	5	6	7	8
0		—	—	0.000 20	0.000 18	0.000 28	0.000 28	0.000 18	0.000 20
1		—	0.007 31	0.005 91	0.004 34	0.004 07	0.004 34	0.005 91	0.007 31
2		0.025 25	0.020 85	0.012 21	0.008 75	0.008 75	0.012 21	0.020 85	0.025 25

Таблица 6.1.4

j	i	1	2	3	4	5	6	7	8
0		—	—	0.001 26	0.001 29	0.002 72	0.003 32	0.002 20	0.002 05
1		—	0.046 03	0.042 78	0.042 08	0.047 38	0.051 91	0.059 50	0.054 68
2		6.292 85	7.306 18	9.817 72	11.971 79	12.405 68	10.259 12	7.601 23	6.702 67

Таблица 6.1.5

j	i	1	2	3	4	5	6	7	8
1		146.726	—206.685	179.969	—159.633	159.633	—179.969	206.685	—146.726
2		—18.440	—10.252	47.315	—29.0123	—29.0123	47.315	—10.252	—18.440

Т а б л и ц а 6.1.6

$j \setminus i$	-1	0	1	2	3	4	5	6	7	8	9	10
0	0.000	0.000	-0.100	-0.500	1.500	2.000	3.000	2.000	1.500	-0.500	-0.100	0.000
1	2.325	2.325	-1.751	0.429	3.457	-1.229	-0.000	1.229	-3.457	-0.429	1.751	-2.325
2	-6.812	-6.812	0.053	9.632	-4.130	0.969	-2.982	0.969	4.130	9.632	0.053	-6.812
3	0.000	0.000	25.741	7.490	-19.497	9.724	0.000	-9.724	19.497	-7.490	-25.741	0.000
4	0.000	0.000	64.354	-155.611	78.504	-25.375	15.650	-25.375	78.504	-155.611	64.354	0.000
5	0.000	80.442	-549.911	334.450	-94.435	20.512	-20.512	94.435	-334.450	549.911	-80.442	0.000

Т а б л и ц а 6.1.7

$j \setminus i$	-1	0	1	2	3	4	5	6	7	8	9	10
0	-5.000	-5.000	-4.500	-4.000	-3.500	-4.000	0.000	4.000	3.500	4.000	4.500	5.000
1	-0.031	-0.031	1.178	1.213	0.067	-0.232	3.506	-0.232	0.067	1.213	1.178	-0.031
2	1.727	1.727	0.865	-0.735	-1.963	1.788	0.000	-1.788	1.963	0.735	-0.865	-1.727
3	0.000	0.000	-3.230	-3.930	0.997	3.007	-2.844	3.007	0.997	-3.930	-3.230	0.000
4	0.000	0.000	-8.075	4.572	9.506	-5.852	0.000	5.852	-9.506	-4.572	8.075	0.000
5	0.000	-10.093	31.617	7.049	-13.962	2.926	2.926	-13.962	7.049	31.617	-10.093	0.000

Т а б л и ц а 6.1.8

x	-0.5	3.5	10.5
$\sigma_1(x)$	-2.013 95	1.648 68	-2.013 95
$\sigma_2(x)$	-4.768 57	-3.844 38	-4.768 57

Таблица 6.1.9

j	i	1	2	3	4	5	6	7	8
0	—	—	—	—	—0.039 30	—0.005 63	—0.001 80	—0.005 63	—0.039 30
1	—	—	—	0.908 15	0.094 15	0.009 89	0.009 89	0.094 15	0.908 15
2	—	—	—9.639 65	—0.961 07	—0.051 87	—0.008 00	—0.051 87	—0.961 07	—9.639 65
3	17.721 51	—	6.471 43	0.296 91	0.032 07	0.032 07	0.296 91	6.471 43	17.721 51

Таблица 6.1.10

j	i	1	2	3	4	5	6	7	8
0	—	—	—	—	—0.009 34	—0.005 08	—0.006 68	—0.038 91	—0.227 39
1	—	—	—	0.215 73	0.065 67	0.028 74	0.064 59	0.536 46	1.834 44
2	—	—	—2.289 87	—0.421 50	—0.082 23	—0.036 69	—0.285 25	—1.752 31	—3.573 30
3	0.237 55	—	0.902 43	3.708 78	6.915 12	5.785 47	2.175 35	0.566 91	0.806 83

Таблица 6.1.11

j	i	1	2	3	4	5	6	7	8
1	—0.643 46	—	—1.640 76	—2.476 45	—1.657 08	1.657 08	2.476 45	1.640 76	0.643 46
2	0.379 79	—	1.039 00	2.829 11	—4.717 01	—4.717 01	2.829 11	1.039 00	0.379 79

Таблица 6.1.12

$i \setminus j$	0	1	2	3	4	5	6	7	8	9	10
x_i	0	0.8	1.2	1.9	3.0	5.0	7.0	8.1	8.8	9.2	10.0
$\sigma_1(x_i)$	-0.353	0.033	0.323	0.945	1.980	2.943	1.980	0.945	0.323	0.033	-0.353
$\sigma_2(x_i)$	-4.792	-4.506	-4.340	-3.989	-3.104	0.000	3.104	3.989	4.340	4.506	4.792

Таблица 6.1.13

$j \setminus i$	-1	0	1	2	3	4	5	6	7	8	9	10
0	-0.353	-0.353	0.033	0.323	0.945	1.980	2.943	1.980	0.945	0.323	0.033	-0.353
1	0.312	0.312	0.649	0.797	0.953	0.866	0.000	-0.866	-0.953	-0.797	-0.649	-0.312
2	0.430	0.430	0.399	0.329	0.103	-0.245	-0.532	-0.245	0.103	0.329	0.399	0.430
3	0.000	0.000	-0.113	-0.243	-0.355	-0.268	0.000	0.268	0.355	0.243	0.113	0.000
4	0.000	0.000	-0.282	-0.370	0.052	0.105	0.162	0.105	0.052	-0.370	-0.282	0.000
5	0.000	-0.353	-0.220	0.603	0.048	0.029	-0.029	-0.048	-0.603	0.220	0.353	0.000

Таблица 6.1.14

$j \setminus i$	-1	0	1	2	3	4	5	6	7	8	9	10
0	-4.792	-4.792	-4.506	-4.340	-3.989	-3.104	0.000	3.104	3.989	4.340	4.506	4.792
1	0.326	0.326	0.392	0.438	0.584	1.079	1.824	1.079	0.584	0.438	0.392	0.326
2	0.078	0.078	0.096	0.138	0.296	0.566	0.000	-0.566	-0.296	-0.138	-0.096	-0.078
3	0.000	0.000	0.067	0.149	0.289	0.076	-0.462	0.076	0.289	0.149	0.067	0.000
4	0.000	0.000	0.167	0.248	0.151	-0.539	0.000	0.539	-0.151	-0.248	-0.167	0.000
5	0.000	0.208	0.203	-0.138	-0.627	0.269	0.269	-0.627	-0.138	-0.203	0.208	0.000

ТАБЛИЦЫ К § 6.2

Таблица 6.2.1

i	0	1	2	3	4	5	6	7	8	9	10
x_i	0.0	0.8	1.2	1.9	3.0	5.0	7.0	8.1	8.8	9.2	10.0
x_{i+1}	-0.05	-0.3	0.5	1.75	2.5	2.5	1.75	0.5	-0.3	-0.05	-
$\int x_i$	-4.75	-4.25	-3.75	-3.75	-4.5	4.5	3.75	3.75	4.25	4.75	-

Таблица 6.2.2

j	i	1	2	3	4	5	6	7	8
0		0.4386	0.4132	0.1462	0.0632	0.0490	0.0849	0.2525	0.4785
1		-0.9171	-0.6657	-0.2311	-0.1123	-0.1123	-0.2311	-0.6657	-0.9171
2		0.4785	0.2525	0.0849	0.0490	0.0632	0.1462	0.4132	0.4386

Таблица 6.2.3

$\begin{matrix} i \\ j \end{matrix}$	1	2	3	4	5	6	7	8
0	—	—	0.000 20	0.000 18	0.000 28	0.000 28	0.000 18	0.000 20
1	—	0.007 31	0.005 91	0.004 34	0.004 07	0.004 34	0.005 91	0.007 31
2	0.025 25	0.020 85	0.012 21	0.008 75	0.008 75	0.012 21	0.020 85	0.025 25

Таблица 6.2.4

$\begin{matrix} i \\ j \end{matrix}$	1	2	3	4	5	6	7	8
0	—	—	0.0013	0.0013	0.0027	0.0033	0.0022	0.0021
1	—	0.0460	0.0428	0.0421	0.0474	0.0519	0.0595	0.0547
2	6.2928	7.3062	9.8177	11.9718	12.4057	10.2591	7.6012	6.7027

Таблица 6.2.5

$\begin{matrix} i \\ j \end{matrix}$	1	2	3	4	5	6	7	8
1	21.3912	—6.3495	—6.5566	—1.2501	—1.2501	—6.5566	—6.3495	21.3912
2	—0.4766	6.0176	—60.2574	156.5836	—156.5836	60.2574	—6.0176	0.4766

Т а б л и ц а 6.2.6

$j \backslash i$	-1	0	1	2	3	4	5	6	7	8	9	10
0	0.3923	0.3923	0.3722	-0.1133	1.1112	2.2092	2.6135	2.2092	1.1112	-0.1133	-0.3722	0.3923
1	-1.2058	-1.2058	-0.2051	1.4535	1.5462	0.5635	0.0000	-0.5635	-1.5462	-1.4535	0.2051	1.2058
2	0.0000	0.0000	3.7528	3.0575	-1.2174	-0.6001	-0.1226	-0.6001	-1.2174	3.0575	3.7528	0.0000
3	0.0000	0.0000	9.3821	-12.8588	0.6448	0.4775	0.0000	-0.4775	-0.6448	12.8588	-9.3821	0.0000
4	0.0000	11.7276	-55.6022	19.2909	-0.1521	-0.2388	-0.2378	-0.1521	19.2909	-55.6022	11.7276	0.0000

Т а б л и ц а 6.2.7

$j \backslash i$	-1	0	1	2	3	4	5	6	7	8	9	10
1	-5.0886	-5.0886	-4.4141	-4.0846	-3.4156	-4.8399	0.0000	4.8399	3.4156	4.0846	4.4141	5.0886
0	0.8488	0.8488	0.8265	0.8543	0.7464	-2.8632	7.5924	-2.8632	0.7464	0.8543	0.8265	0.8488
2	0.0000	0.0000	-0.0836	0.4175	-2.2476	0.3320	0.0000	-3.3201	2.2476	-0.4175	0.0836	0.0000
3	0.0000	0.0000	-0.2090	2.7148	-10.3292	15.0193	-15.3513	15.0193	-10.3292	2.7146	-0.2090	0.0000
4	0.0000	-0.2613	7.3091	-18.6340	23.0441	-15.1853	15.1853	-23.0441	18.6340	-7.3091	0.2613	0.0000

Т а б л и ц а 6.2.8

x	-0.5	3.5	10.5
$\sigma_1(x)$	0.9952	2.4252	0.9952
$\sigma_2(x)$	-5.5129	-5.9566	5.5129

Таблица 6.2.9

$j \backslash i$	1	2	3	4	5	6	7	8
0	—	—	0.0701	0.0161	0.0044	0.0044	0.0161	0.0701
1	—	0.6902	—0.1498	—0.0198	—0.0069	—0.0198	—0.1498	—0.6902
2	1.2876	0.6986	0.0942	0.0278	0.0278	0.0942	0.6986	1.2876

Таблица 6.2.10

$j \backslash i$	1	2	3	4	5	6	7	8
0	—	—	0.0618	0.0282	0.0194	0.0292	0.1002	0.2482
1	—	—0.6082	—0.1957	—0.0626	—0.0377	—0.1161	—0.4888	—0.8483
2	0.8813	1.7444	4.3819	6.5878	6.2063	3.5387	1.4914	1.4054

Таблица 6.2.11

$j \backslash i$	1	2	3	4	5	6	7	8
1	0.6464	0.2981	—1.9112	—3.9197	—3.9197	—1.9112	0.2981	0.6464
2	—0.2990	—0.2367	3.4433	16.6004	—16.6004	—3.4433	0.2367	0.2990

Т а б л и ц а 6.2.12

$j \backslash i$	-1	0	1	2	3	4	5	6	7	8	9	10
0	-0.6605	-0.6605	-0.0029	0.3458	0.9954	1.9248	2.6648	1.9248	0.9954	0.3458	-0.0029	-0.6605
1	0.8144	0.8144	0.8447	0.9002	0.9338	0.7114	0.0000	-0.7114	-0.9338	-0.9002	0.8447	-0.8144
2	0.0000	0.0000	0.1134	0.1329	-0.0564	-0.2986	-0.3843	-0.2986	-0.0564	0.1329	0.1134	0.0000
3	0.0000	0.0000	0.2835	-0.1861	-0.3547	-0.0857	0.0000	0.0857	0.3547	0.1861	-0.2835	0.0000
4	0.0000	0.3544	-1.1739	-0.2408	0.2445	-0.0428	0.0428	0.2445	-0.2408	-1.1739	0.3544	0.0000

Т а б л и ц а 6.2.13

$j \backslash i$	-1	0	1	2	3	4	5	6	7	8	9	10
0	-4.7511	-4.7511	-4.4883	-4.3663	-4.1695	-3.6217	0.0000	3.6217	4.1695	4.3663	4.4883	4.7511
1	0.3319	0.3319	0.3180	0.2912	0.2958	0.8334	2.4983	0.8334	0.2958	0.2912	0.3180	0.3319
2	0.0000	0.0000	-0.0525	-0.0696	0.1433	0.8698	0.0000	-0.8698	-0.1433	0.0696	0.0525	0.0000
3	0.0000	0.0000	-0.1311	0.0453	0.5632	0.7577	-1.6275	0.7577	0.5632	0.0453	-0.0453	0.0000
4	0.0000	-0.1639	0.4410	0.7399	0.1768	-1.1926	1.1926	-0.1768	-0.7399	-0.4410	0.1639	0.0000

Т а б л и ц а 6.2.14

x	-0.5	3.5	10.5
$\sigma_1(x)$	-1.0677	2.2415	-1.0677
$\sigma_2(x)$	-4.9170	-3.0835	4.9170

ТАБЛИЦЫ К § 6.3

Т а б л и ц а 6.3.1

i	1	2	3	4	5
x_i	$-2/3$	0.25	-0.40	0.00	0.70
y_i	$-1/3$	-0.50	0.70	0.00	-0.40

Т а б л и ц а 6.3.2

R_0	1	2	3	4	5	6	7	8
0.00	0.2907	-0.6549	-0.0556	-0.0593	0.4790	0.9345	-0.9347	-0.8489
5.00	0.1935	-0.4129	-0.0196	-0.0815	0.3206	0.9902	-1.0106	-0.8694

Т а б л и ц а 6.3.3

j	1	2
1	0	0
2	1	0
3	0	1

Т а б л и ц а 6.3.4

j	1	2	3	4	5
1	0.0000	0.6707	0.1951	0.4878	1.0000
2	0.2439	0.1220	1.0000	0.4878	0.1951

Т а б л и ц а 6.3.5

i	1	2	3	4	5
R_i	-1	1	-1	1	-1

Т а б л и ц а 6.3.6

R_0	1	2	3	4	5	6	7	8
0.00	-1.4284	2.4651	-0.2967	1.6704	-2.4104	0.5189	-0.0949	-0.5019
5.00	-0.9889	1.5901	-0.2938	1.3703	-1.6776	0.2925	0.1356	-0.4877

Т а б л и ц а 6.3.7

R	$\sigma(P_1)$	$\sigma(P_2)$
0.00	-1.0000	-0.8654
5.00	-0.8022	-0.8125

Т а б л и ц а 6.3.8

i	1	2	3	4	5
x_i	-2/3	0.25	-0.40	0.00	0.70
y_i	-1/3	-0.50	0.70	0.00	-0.40
z_i	1/3	0.50	-0.20	0.30	0.40

Т а б л и ц а 6.3.9

R_0	j	1	2	3	4	5	6	7	8	9
0.00		-0.2071	0.6580	0.1832	-0.3054	-0.3268	3.9143	-0.9050	-2.5576	-3.2943
5.00		-0.4230	0.3898	0.1089	-0.1815	-0.1942	3.8796	-0.9460	-2.5411	-3.2195

Т а б л и ц а 6.3.10

Т а б л и ц а 6.3.11

j	i	1	2	3	4	5	j	i	1	2	3
1		0.0000	0.6707	0.1951	0.4878	1.0000	1		0	0	0
2		0.2439	0.1220	1.0000	0.4878	0.1951	2		1	0	0
3		0.7317	0.8537	0.3415	0.7073	0.7805	3		0	1	0
							4		0	0	1

Т а б л и ц а 6.3.12

R_0	i	1	2	3	4	5	6	7	8	9
0.00		0.7012	-2.2214	-0.6205	1.0342	1.1066	-21.430	-0.8057	12.048	24.460
5.00		0.4167	-1.3201	-0.3688	0.6146	0.6576	-21.312	-0.6666	11.992	24.206

Т а б л и ц а 6.3.13

Т а б л и ц а 6.3.14

R_0	$\sigma(P_1)$	$\sigma(P_2)$	Режим				
			j	1	8	16	25
0.00	—1.0000	9.4893	Интерполяция Сглаживание	1.2146	—0.1703	0.2712	6.7971
5.00	—0.9167	9.4196		1.0274	—0.1509	0.1655	6.7333

Т а б л и ц а 6.3.15

Т а б л и ц а 6.3.16

Режим	i	1	6	8	25	Режим	$\sigma(P_1)$	$\sigma(P_2)$
Интерполяция		—7.1172	—2.0935	0.1700	8.1334	Интерполяция	—1.0000	—0.9367
Сглаживание		—5.2618	—1.3893	0.4789	0.9952	Сглаживание	—0.9474	—0.8582

Т а б л и ц а 6.3.17

i	1	2	3	4	5	6	7	8
x_i	—2/3	0.25	—0.40	0.00	0.70	0.30	0.15	—0.25
y_i	—1/3	—0.50	0.70	0.00	—0.40	—0.40	0.30	—0.25
z_i	1/3	0.50	—0.20	0.30	0.40	0.10	—0.25	—0.40
f_i	—1	1	—1	1	—1	1	—1	1

Таблица 6.3.18

j	i	2	10	21	40
2		0.0000	0.0000	0.0000	5.9209
5		0.0000	0.0000	0.0000	—58.157
19		0.0000	0.0000	—0.0018	—0.5905
40		5.9209	1.2742	0.5624	187.017

Таблица 6.3.19

i	1	8	16	40	44
$B[i]$	0.0000	0.0000	0.0000	107.80	—274.30

К главе 1

Понятие сплайна было впервые введено в работе И. Дж. Шёнберга [61]. Теория полиномиальных сплайнов развивалась в работах И. Дж. Шёнберга, Дж. Алберга, Э. Нилсона, Дж. Уолша, Г. Биргофа, Р. С. Варги, Т. Гревилля, Де Бура, А. Сарда, Ю. С. Завьялова, С. Б. Стечкина, Ю. Н. Субботина, Н. П. Корнейчука, В. А. Морозова и многих других авторов. Экстремальное свойство сплайнов, позволившее дать их вариационную формулировку, было открыто Дж. Холлидеем [60]. Понятие сплайна как решения вариационной задачи в абстрактном гильбертовом пространстве дано М. Атья [53]. Теоремы существования и единственности таких сплайнов доказаны Ф. М. Анселоном и П.-Ж. Лораном [52].

К главе 2

Общие алгоритмы для построения интерполяционных и сглаживающих сплайнов были предложены Ф. М. Анселоном и П.-Ж. Лораном [52]. Для случая сглаживающих сплайнов алгоритм был модифицирован в работе А. З. Ананьина, В. В. Смелова, В. А. Василенко [15]. Применение этих алгоритмов, однако, нетривиально даже в случае кусочно-полиномиальных сплайнов высоких степеней. Это связано, во-первых, с трудностью обращения неявно заданного оператора T^* (вопрос решен в п. 2.3.2), во-вторых, с обеспечением устойчивости вычисления сплайна по известной производной. Идея устойчивого алгоритма изложена П.-Ж. Лораном в [8]. Полиномиальные сплайны нескольких переменных как результат решения серии одномерных задач изучались в работах Ю. С. Завьялова [27, 28, 29], Дж. Алберга, Э. Нильсона, Дж. Уолша [1, 48—51]. Соответствующая вариационная формулировка таких задач дана в работах Ю. С. Завьялова, А. Имамова [32, 33]. Алгоритм построения аналитического сплай-

на на хаотической сетке был предложен Ю. С. Завьяловым и А. Имамовым [33], для случая наклонов, кривизн, интегральных средних обобщен в работе А. В. Ковалкова [40]. В книге эти результаты изложены в более общей форме.

К главе 3

Метод конечных элементов для приближенного вычисления сложных неполиномиальных сплайн-функций, в частности на хаотических сетках, был предложен и обоснован в работах автора [3, 19, 20].

К главе 4

Общая теория сходимости сплайнов на основе концепции правильных систем операторов была развита автором в работах [16, 17, 3]. На основе общих теорем им были получены оценки сходимости сплайнов на сгущающихся хаотических сетках [3]. В дальнейшем А. Имамов доказал [37, 38, 39], что свойство «правильности» является необходимым для сходимости интерполяционного процесса. Сходимость сглаживающих сплайнов к интерполяционным была доказана В. А. Морозовым [44]. Проблема выбора параметра сглаживания изучена в работах К. Райнша [57, 58], В. А. Морозова, В. И. Гордоновой [45, 46].

К главе 5

Основу этой главы составили работы автора совместно с М. В. Зюзиным [23, 24, 25].

Вопросам построения цифровых фильтров посвящена обширная литература как математического, так и инженерного характера. Мы прежде всего рекомендуем читателю превосходную монографию [47], в которой удачно сочетается полнота и доступность изложения соответствующих вопросов.

К главе 6

Библиотека программ *LIDA* была создана в лаборатории численных методов анализа Вычислительного центра СО АН СССР г. Новосибирска. Руководителем работ был автор; комплект *ODD* написан автором. *EVEN* — автором и Г. С. Беленьким, *GREEN* — А. В. Ковалковым, комплект *AVERAGE* — автором и М. В. Зюзиным. Инструктивный материал к библиотеке *LIDA* изложен В. А. Василенко, А. В. Ковалковым и М. В. Зюзиным в [26].

1. Альберг Дж., Нильсон Э., Уолш Дж. Теория сплайнов и ее приложения. М.: Мир, 1972. 316 с.
2. Бердышев В. И., Субботин Ю. Н. Численные методы приближенных функций. Свердловск: Среднеуральское кн. изд-во, 1979. 120 с.
3. Василенко В. А. Теория сплайн-функций. Новосибирск: НГУ, 1978. 65 с.
4. Гахов Д. Ф., Черский Ю. И. Уравнения типа свертки. М.: Наука, 1978. 295 с.
5. Гребенников А. И. Метод сплайнов в численном анализе. М.: Изд-во МГУ, 1979. 99 с.
6. Завьялов Ю. С., Квасов Б. И., Мирошниченко В. Л. Методы сплайн-функций. М.: Наука, 1980. 352 с.
7. Лионс Ж.-Л., Мадженес Э. Неоднородные граничные задачи и их приложения. М.: Мир, 1971. 372 с.
8. Лоран П.-Ж. Аппроксимация и оптимизация. М.: Мир, 1975. 496 с.
9. Люстерник Л. А., Соболев В. Н. Элементы функционального анализа. М.: Наука, 1965. 519 с.
10. Марчук Г. И. Методы вычислительной математики. М.: Наука, 1980. 534 с.
11. Марчук Г. И., Кузнецов Ю. А. Итерационные методы и квадратичные функционалы.— В кн.: Методы вычислительной математики. Новосибирск: Наука, 1975, с. 4—143.
12. Стечкин С. Б., Субботин Ю. Н. Сплайны в вычислительной математике. М.: Наука, 1976. 248 с.
13. Тихонов А. Н., Арсенин В. Я. Методы решения некорректных задач. М.: Наука, 1974. 223 с.
14. Уилкинсон Дж., Райнш К. Справочник алгоритмов на языке АЛГОЛ. Линейная алгебра. М.: Машиностроение, 1976. 389 с.
15. Ананьин А. З., Смелов В. В., Василенко В. А. Эффективный способ преобразования вариационной задачи сглаживания к линейной алгебраической системе.— В кн.: Дифференциальные и интегродифференциальные уравнения. Новосибирск: ВЦ СО АН СССР, 1977, с. 127—135.
16. Василенко В. А. Сходимость сплайнов в гильбертовом пространстве.— Числ. методы механики сплош. среды, Новосибирск, 1972, т. 3, № 3, с. 18—23.
17. Василенко В. А. Сходимость операторных интерполирующих сплайнов.— В кн.: Вариационно-разностные методы в математической физике. Новосибирск: ВЦ СО АН СССР, 1973, с. 95—100.

18. Василенко В. А. Сглаживание операторными сплайнами.— В кн.: Вариационно-разностные методы в математической физике. Новосибирск: ВЦ СО АН СССР, 1973, с. 101—106.
19. Василенко В. А. Сглаживающие сплайны на подпространствах и теоремы компактности.— Числ. методы механики сплош. среды, Новосибирск, 1974, т. 5, № 5, с. 37—42.
20. Василенко В. А. Конечномерная аппроксимация в методе наименьших квадратов.— В кн.: Вариационно-разностные методы в математической физике. Новосибирск: ВЦ СО АН СССР, с. 160—172.
21. Василенко В. А. Избыточная гладкость сплайн-интерполянтов.— Вычислительные системы, Новосибирск, 1977, № 72, с. 3—9.
22. Василенко В. А. Приближенное решение задачи продолжения функций методом конечных элементов.— В кн.: Вариационно-разностные методы в математической физике. Новосибирск: ВЦ СО АН СССР, 1978, с. 142—149.
23. Василенко В. А., Зюзин М. В. О применении осредняющих функций в задачах обработки экспериментальных данных. Препринт № 59. Новосибирск: ВЦ СО АН СССР, 1979. 18 с.
24. Василенко В. А., Зюзин М. В. Обработка экспериментальных данных с применением осреднений, сохраняющих полиномиальные тригонометрические и экспоненциальные функции.— В кн.: Теория кубатурных формул и вычислительная математика. Новосибирск: Наука, 1980, с. 131—135.
25. Василенко В. А., Зюзин М. В. Осредняющие операторы типа свертки и обработка экспериментальных данных.— В кн.: Вариационно-разностные методы в математической физике. Новосибирск: ВЦ СО АН СССР, 1981, с. 36—46.
26. Василенко В. А., Ковалков А. В., Зюзин М. В. Библиотека программ «LIDA» для аппроксимации функций и обработки данных. Вариант АЛГОЛ — БЭСМ-6. Препринт № 270. Новосибирск: ВЦ СО АН СССР, 1981. 37 с.
27. Завьялов Ю. С. Интерполирование функций одной из двух переменных кусочно-полиномиальными функциями.— В кн.: Математические проблемы геофизики. Новосибирск: ВЦ СО АН СССР, 1969, ч. I, с. 125—141.
28. Завьялов Ю. С. Интерполирование кубическими многозвенниками (сплайнами).— Вычислительные системы, Новосибирск, 1970, № 38, с. 23—73.
29. Завьялов Ю. С. Интерполирование бикубическими многозвенниками (сплайнами).— Вычислительные системы, Новосибирск, 1970, № 38, с. 74—101.
30. Завьялов Ю. С. Экстремальное свойство кубических многозвенников (сплайнов) и задача сглаживания.— Вычислительные системы, Новосибирск, 1970, № 42, с. 89—108.
31. Завьялов Ю. С. L — сплайн-функции многих переменных.— ДАН СССР, 1974, т. 214, № 6, с. 1247—1249.
32. Завьялов Ю. С., Имамов А. Алгоритмы с расщеплением решения задачи сглаживания сплайн-функциями многих переменных.— Числ. методы механики сплош. среды, Новосибирск, 1976, т. 7, № 6, с. 52—61.
33. Завьялов Ю. С., Имамов А. О вариационных задачах теории сплайнов.— В кн.: Математический анализ и смежные вопросы математики. Новосибирск: Наука, 1978, с. 27—36.

34. Имамов А. О некоторых экстремальных свойствах сплайнов многих переменных.— Вычислительные системы, Новосибирск, 1975, № 65, с. 68—73.
35. Имамов А. Приближенное решение задачи сглаживания.— Вычислительные системы, Новосибирск, 1975, № 65, с. 74—82.
36. Имамов А. О сглаживающих сплайнах.— Вычислительные системы, Новосибирск, 1978, № 75, с. 3—15.
37. Имамов А. Сходимость интерполяционного процесса в гильбертовом пространстве и ее применения.— Числ. методы механики сплош. среды, Новосибирск, 1976, т. 7, № 7, с. 15—21.
38. Имамов А. Некоторые вопросы теории сплайнов в гильбертовом пространстве. Автореф. канд. дис. Новосибирск: ВЦ СО АН СССР, 1977. 8 с.
39. Имамов А. Сходимость интерполяционного процесса в гильбертовом пространстве и ее применения.— Числ. методы механики сплош. среды, Новосибирск, 1977, т. 8, № 1, с. 49—58.
40. Ковалков А. В. Функции Грина и сплайн-аппроксимация в многомерных областях. Препринт № 70. Новосибирск: ВЦ СО АН СССР, 1980. 21 с.
41. Мирошниченко В. Л. Об интерполировании кубическими сплайнами.— Вычислительные системы, Новосибирск, 1973, № 56, с. 18—22.
42. Мирошниченко В. Л. Интерполяция функций с большими градиентами.— В кн.: Методы аппроксимации и интерполяции. Новосибирск: ВЦ СО АН СССР, 1981, с. 27—35.
43. Морозов В. А. Теория сплайнов и задача устойчивого вычисления значений неограниченного оператора.— Журн. вычисл. математики и мат. физики, 1971, т. 11, № 3, с. 545—548.
44. Морозов В. А. О приближенном решении операторных уравнений методом сплайнов.— ДАН СССР, 1971, т. 200, № 1, с. 35—39.
45. Морозов В. А. О выборе параметра при решении функциональных уравнений методом регуляризации.— ДАН СССР, 1967, т. 175, № 6, с. 1225—1228.
46. Гордонова В. И., Морозов В. А. Численные алгоритмы выбора параметра в методе регуляризации.— Журн. вычисл. математики и мат. физики, 1973, т. 13, № 3, с. 539—545.
47. Рабинер Л., Голд Б. Теория и применение цифровой обработки сигналов. М.: Мир, 1978. 848 с.
48. Альберг Дж., Нильсон Э., Уолш Дж. (Ahlberg J. H., Nilson E. N., Walsh J. L.) Orthogonality properties of the spline function.— J. Math. Anal. Appl., 1965, v. 11, p. 321—337.
49. Альберг Дж., Нильсон Э., Уолш Дж. (Ahlberg J. H., Nilson E. N., Walsh J. L.) Fundamental properties of generalized splines.— Proc. Nat. Acad. Sci. USA, 1964, v. 52, p. 1412—1419.
50. Альберг Дж., Нильсон Э., Уолш Дж. (Ahlberg J. H., Nilson E. N., Walsh J. L.) Best approximation and convergence properties of higher order spline approximations.— J. Math. Mech., 1965, v. 14, p. 231—244.
51. Альберг Дж., Нильсон Э., Уолш Дж. (Ahlberg J. H., Nilson E. N., Walsh J. L.) Extremal orthogonality and convergence of multidimensional splines.— J. Math. Appl., 1965, v. 11, p. 27—48.
52. Анселон Ф. М., Лоран П.-Ж. (Anselone P. M., Laurent P.-J.) A general method for the construction of interpolating or smoothing spline functions.— Numer. Math., 1968, v. 12, N 1, p. 66—82.

53. Атья М. (Atteia M.) Généralisation de la definition et des propriétés des "spline function".—Comp. Rend., 1965, v. 260, p. 3550—3553.
54. Биркгоф Г., Де Бур К. (Birkhoff G., de Boor C.) Error bounds for spline interpolation.—J. Math. Mech., 1964, vol. 13, p. 827—835.
55. Де Бур К. (de Boor C.) Best approximation properties of spline functions of odd degree.—J. Math. Mech., 1963, v. 12, p. 747—749.
56. Гревилль Т. (Greville T. N. E.) Numerical procedures for interpolation by spline functions.—J. Soc. Ind. Appl. Math. Numer. Anal., 1964, ser. B, v. 1, p. 53—68.
57. Райнш К. (Reinsch C. H.) Smoothing by spline functions.—Numer. Math., 1967, v. 10, p. 177—183.
58. Райнш К. (Reinsch C. H.) Smoothing by spline functions II.—Numer. Math., 1971, v. 16, p. 451—454.
59. Сард А. (Sard A.) Optimal approximation.—J. Funct. Anal., 1967, v. 1, p. 222—244.
60. Холлидей Дж. (Holladay J. C.) Smoothest curve approximation.—Math. Tables Aids Comput., 1957, v. 11, p. 233—243.
61. Шёнберг И. Дж. (Schoenberg I. J.) Contributions to problem of approximation of equidistant data by analytic functions. Quart.—Appl. Math., 1946, v. 4, p. 45—99, 112—141.
62. Шёнберг И. Дж. (Schoenberg I. J.) On Polya frequency functions and their Laplace transforms.—J. Anal. Math., 1951, v. 1, p. 331—374.
63. Шёнберг И. Дж. (Schoenberg I. J.) On trigonometric spline interpolation.—J. Math. Mech., 1964, v. 13, p. 795—825.

ОГЛАВЛЕНИЕ

Предисловие	3
Введение	7
Глава 1. Основные понятия теории сплайнов	10
§ 1.1. Понятие интерполяционных и сглаживающих сплайнов	—
§ 1.2. Теоремы существования и единственности	12
§ 1.3. Примеры задач интерполяции и сглаживания	17
Глава 2. Аналитические методы построения сплайн-функций	22
§ 2.1. Общий алгоритм построения интерполяционного сплайна	—
§ 2.2. Общий алгоритм построения сглаживающего сплайна	25
§ 2.3. Сплайны нечетной степени на отрезке	27
2.3.1. Формирование матрицы H	28
2.3.2. Обращение оператора T^*	29
2.3.3. Формирование матрицы A	30
2.3.4. Решение системы	33
2.3.5. Вычисление полиномиального представления	34
§ 2.4. Сплайны четной степени на отрезке и аппроксимация по локальным интегралам	36
2.4.1. Формирование матрицы H	38
2.4.2. Формирование матрицы A	39
2.4.3. Решение системы и вычисление полиномиального представления	40
§ 2.5. Сплайн-интерполяция в областях типа параллелепипеда	42
§ 2.6. Функции Грина и аналитическое представление сплайнов в многомерных областях произвольной конфигурации	45
§ 2.7. Сплайн-аппроксимация в областях с хаотически расположенными интерполяционными узлами	48
Глава 3. Метод конечных элементов при построении сплайн-функций	52
§ 3.1. Интерполяционные сплайны на подпространствах	53

§ 3.2. Сглаживающие сплайны на подпространствах	58
§ 3.3. Метод конечных элементов на примере двумерной задачи сплайн-аппроксимации	61
Глава 4. Сходимость сплайн-функций	65
§ 4.1. Основная теорема сходимости	—
§ 4.2. Сходимость сплайнов нечетной степени на отрезке	68
§ 4.3. Сходимость сплайнов в двумерной области на сгущающейся хаотической сетке	70
§ 4.4. Сходимость сглаживающих сплайнов к интерполяционным	74
§ 4.5. Выбор параметра сглаживания	78
Глава 5. Применение осредняющих операторов типа свертки для обработки экспериментальных данных	82
§ 5.1. Некоторые свойства осредняющих операторов типа свертки	83
§ 5.2. Осредняющие функции, линейно зависящие от конечного числа параметров	85
§ 5.3. Кусочно-постоянные осредняющие функции	88
§ 5.4. Осреднения, сохраняющие тригонометрические функции наперед заданных частот	90
§ 5.5. Осреднения, сохраняющие экспоненциальные функции заданных показателей роста или убывания	93
§ 5.6. Осреднения, сохраняющие полиномы заданной степени	94
§ 5.7. Что происходит, если параметры информативной части кривой заданы неточно?	96
§ 5.8. Вычислительный процесс	98
Глава 6. Описание библиотеки программ аппроксимации функций <i>LIDA</i>	101
§ 6.1. Комплект <i>ODD</i> для расчета интерполяционных и сглаживающих сплайнов нечетной степени на отрезке	102
6.1.1. Процедура <i>ASSH</i>	—
6.1.2. Процедура <i>ASSA</i>	103
6.1.3. Процедура <i>DECOMB</i>	104
6.1.4. Процедура <i>SOLVE</i>	—
6.1.5. Процедура <i>REPRES</i>	105
6.1.6. Вещественная процедура <i>FN</i>	107
6.1.7. Процедура выбора параметра сглаживания <i>BESPAR</i>	—
6.1.8. Общие замечания по работе комплекта <i>ODD</i>	108
6.1.9. Тест режима интерполяции	109
6.1.10. Тест режима сглаживания	—
6.1.11. Тест процедуры <i>BESPAR</i>	110
§ 6.2. Комплект <i>EVEN</i> для расчета интерполяционных и сглаживающих сплайнов четной степени на отрезке по заданным локальным интегралам	—
6.2.1. Процедура <i>ASSH2</i>	111
6.2.2. Процедура <i>ASSA2</i>	—

6.2.3.	Процедура <i>DECOMB2</i>	111
6.2.4.	Процедура <i>SOLVE2</i>	112
6.2.5.	Процедура <i>REPRES2</i>	—
6.2.6.	Вещественная процедура-функция <i>FN2</i>	113
6.2.7.	Процедура выбора параметра сглаживания <i>BESPAR2</i>	—
6.2.8.	Общие замечания по работе с комплектом <i>EVEN</i>	114
6.2.9.	Тест режима интерполяции	115
6.2.10.	Тест режима сглаживания	—
6.2.11.	Тест процедуры <i>BESPAR2</i>	—
§ 6.3.	Комплект <i>GREEN</i> для сплайн-аппроксимации в многомерных областях произвольной конфигурации с хаотическими сетками	116
6.3.1.	Процедура <i>FORMH</i>	117
6.3.2.	Процедура <i>SOL</i>	118
6.3.3.	Процедура <i>VAL</i>	119
6.3.4.	Процедура <i>FORMH2</i>	120
6.3.5.	Процедура <i>SOL2</i>	121
6.3.6.	Процедура-функция <i>VAL2</i>	123
6.3.7.	Рекомендуемый режим работы с комплектом <i>GREEN</i>	—
6.3.8.	Качественный тест комплекта <i>GREEN</i>	124
6.3.9.	Отладочный тест процедур <i>FORMH</i> , <i>SOL</i> , <i>VAL</i>	125
6.3.10.	Отладочный тест процедур <i>FORMH2</i> , <i>SOL2</i> , <i>VAL2</i>	126
6.3.11.	Процедура выбора параметра сглаживания <i>BESFOR</i>	127
6.3.12.	Тест процедуры <i>BESFOR</i>	128
6.3.13.	Процедура выбора параметров сглаживания <i>BESFO2</i>	129
6.3.14.	Тест процедуры <i>BESFO2</i>	130
§ 6.4.	Комплект <i>AVERAGE</i> для фильтрации зашумленных данных	131
6.4.1.	Описание комплекта <i>AVERAGE</i>	—
6.4.2.	Пример работы с комплектом <i>AVERAGE</i> и отладочный тест	133
Приложение 1.	Текст комплекта <i>ODD</i>	136
Приложение 2.	Текст комплекта <i>EVEN</i>	144
Приложение 3.	Общая часть <i>ODD</i> и <i>EVEN</i>	152
Приложение 4.	Текст комплекта <i>GREEN</i>	155
Приложение 5.	Текст комплекта <i>AVERAGE</i>	186
Приложение 6.	Таблицы к § 6.1	191
Приложение 7.	Таблицы к § 6.2	196
Приложение 8.	Таблицы к § 6.3	201
Комментарии		206
Литература		208

Владимир Александрович Василенко

**СПЛАЙН-ФУНКЦИИ:
ТЕОРИЯ, АЛГОРИТМЫ, ПРОГРАММЫ**

Ответственный редактор
Гурий Иванович Марчук

Утверждено к печати
Вычислительным центром СО АН СССР

Редактор издательства *Т. Д. Семченко*
Художественный редактор *Т. Ф. Каминина*
Художник *Н. А. Пискун*
Технический редактор *Н. М. Остроумова*
Корректоры *Е. Н. Зими́на, Н. И. Молосто́ва*

ИБ № 23197

Сдано в набор 21.05.82. Подписано к печати 25.04.83. МН 07525. Формат 84×108^{1/32}. Бумага типографская № 2. Обыкновенная гарнитура. Высокая печать. Усл. печ. л. 11,3. Усл. кр.-отт. 11,5. Уч.-изд. л. 14. Тираж 5000 экз. Заказ № 182. Цена 1 р. 40 к.

Издательство «Наука», Сибирское отделение.
630090, Новосибирск, 99, Советская, 18.
Типография издательства «Наука».
630077, Новосибирск, 77, Станиславского, 25.